

测试度量指标介绍

发布时间：2007-9-06 15:52 作者：Judy Shen 来源：Judy Shen 的博客

在 CMMI4 体系的测试过程中定义了四个度量指标：测试覆盖率、测试执行率、测试执行通过率、测试缺陷解决率。为了使专/兼职测试人员理解这四个度量指标，了解如何利用现有资源收集度量数据，本文介绍这四个指标的含义及数据收集方法。

1、测试覆盖率

测试覆盖率是指测试用例对需求的覆盖情况。

计算公式：已设计测试用例的需求数/需求总数。

测试覆盖率从纬度上说包括广度覆盖和深度覆盖；从内容上说包括用户场景覆盖、功能覆盖、功能组合覆盖、系统场景覆盖。

首先说广度，是否需求规格说明书中的每个需求项都在测试用例中得到设计。其次说深度，通俗的说，是不使我们的测试设计流于表面，是否能够透过客户需求文档，挖掘出可能存在问题的地方。例如：重复点击某个按钮 10 次，或者依次执行新增、删除、新增同一数据的记录、再次删除该记录操作。在笔者的实际工作中碰到过这么一个例子，一个使用 PL/SQL 编写的系统，在某个查询界面，重复点击《查询》按钮 6 次后，系统就会出现查询功能失效的问题。经调试，开发人员发现是由于 gdi 资源未完全释放的缘故。

在设计测试用例时，我们很少单独设计广度或深度方面的测试用例，而一般是结合在一起设计。为了从广度和深度上覆盖测试用例，我们需要考虑设计各种测试用例，如：用户场景（识别最常用的 20%的操作）、功能点、功能组合、系统场景、性能、语句、分支等。在执行时，需要根据测试时间的充裕程度按照一定的顺序执行。通常是先执行用户场景的测试用例，然后再执行具体功能点、功能组合的测试。

测试覆盖率数据的收集，我们可以通过需求跟踪矩阵RTM来实现。在需求跟踪矩阵，测试人员填写的“系统测试用例”列的数据，如图一所示。测试人员通过计算RTM列出的需求数量，和已设计测试用例的需求数量，可以快速的计算出测试覆盖率。通过RTM，测试人员，包括项目组成员都可以很清楚的、快速的知

道当前这个项目测试的测试覆盖情况。

质量管理体系					
需求跟踪矩阵					
57	57	57	57	57	4
需求号	软件需求详述	需求状态	需求类型	优先级	系统测试用例
1.1	用户管理	初稿	功能性	强制的	TC_INN_UserMgr
1.2	基础数据管理	初稿	功能性	强制的	TC_INN_OrgMgr
					TC_INN_EquMgr
					TC_INN_EquSerMgr
					TC_INN_EquSanMgr
1.3	运行参数管理	初稿	功能性	强制的	TC_INN_RunParaMgr
1.4	数据导入导出	初稿	功能性	强制的	TC_INN_DateInp
					TC_INN_DateExp
					TC_INN_ExpBaseDate
					TC_INN_InpBaseDate
					TC_INN_ExpATMHistTra
					TC_INN_InpATMHistTra

图一 需求跟踪矩阵例子

注：本 RTM 例子中，笔者将“概要设计”、“详细设计”、“编码”等列隐藏，只显示与测试覆盖率计算有关的内容。

2、测试执行率

执行率，顾名思义，就是指实际执行过程中确定已经执行的测试用例比率。
计算公式：已执行的测试用例数/设计的总测试用例数。

读者肯定觉得很奇怪了，我们设计的测试用例肯定都是要执行的，即使是按模块来执行测试，那该模块的测试执行率肯定是 100%，为什么还要设置这个指标？

其实不然。在实际测试过程中，经常有如下这种情况发生。一种情况是，因为系统采用迭代方式开发，每次 Build 时都有不同的重点，包含不同的内容；第二种情况是，由于测试资源的有限，不可能每次将所有设计的测试内容都全部测试完毕。由于这两种情况的存在，所以在每次执行测试时，我们会按照不同的测试重点和测试内容来安排测试活动，所以就存在了“测试执行率”这个指标。

通常，我们的测试目标是确保 100%的测试用例都得到执行，即执行率为 100%。但是，如前面所提到的，实际中可能存在非 100%的执行率。如果不能达到 100%的测试执行率，那么我们需要根据不同的情况制定不同的测试执行率标准——主要考虑风险、重要性、可接受的测试执行率。在考虑可接受的测试执行

率时，就涉及到了测试用例执行顺序的问题。

在设计测试用例时，我们需要从广度和深度上尽可能的覆盖需求，所以我们就需要设计各种测试用例，如正常的测试用例、异常的测试用例、界面的测试用例等。但是在执行时，测试人员需要根据项目进度和测试时间的充裕程度，参考测试执行率标准，将测试用例按照一定的顺序执行。通常是先执行用户场景对应的测试用例，然后再执行具体功能点、功能组合的测试，完成这些测试后，再进行其它测试，如系统场景、性能、语句等测试。

例如，某项目共设计了 280 个测试用例。该项目某一阶段的测试共分四个版本，其中有一个版本执行了 134 个测试用例，那么该版本的测试执行率为 47.9%。

3、测试执行通过率

在介绍“测试执行通过率”之前，需要说明测试用例的执行结果定义。测试用例的执行结果有以下四种定义：

执行结果	说明
通过Pass	实际结果与预期结果一致，或者有一些不同，但是影响不大，可以接受。
失败Fail	实际结果与预期结果不一致，存在缺陷。
阻塞Block	因为前面测试用例执行失败，导致本测试用例无法执行。
忽略Skip	测试用例不适用。存在这类测试用例时，应调整测试用例，使其符合新的系统需求。

介绍了执行结果定义后，我们来看测试执行通过率。测试执行通过率，指在实际执行的测试用例中，执行结果为“通过”的测试用例比率。

计算公式：执行结果为“通过”的测试用例数/实际执行的测试用例总数。

我们可以针对所有计划执行的测试用例进行衡量，可以细化到具体模块，用于对比各个模块的测试用例执行情况。

为了得到测试执行通过率数据，我们在测试执行时，需要在测试用例副本中记录下每个测试用例的执行结果，然后在当前版本执行完毕，或者定期（如每周）统计当前测试执行数据。通过原始数据的记录与统计，我们可以快速的得到当前版本或当前阶段的测试执行通过率。

下表是某项目某一测试版本使用测试用例执行测试的统计数据：

模块	测试执行结果	通过	失败	阻塞	忽略
	系统用例				
会员管理	报名参赛（UC1）	12	0	0	4
	赛事查询（UC2）	5	0	0	0
	比赛资格查询（UC3）	4	0	0	3
	加入战队（UC4）	18	0	0	0
	组织我的战队（UC5）	29	0	0	0
	查询我的战队（UC10）	5	0	0	0
	会员注册	11	0	0	0
	修改会员资料	9	0	0	0
基本信息管理	比赛项目管理（UC6）	12	2	0	1
	赛区运营商（UC10）	8	1	0	0
	赛场管理（包括支持项目）（UC8）	16	0	0	0
赛事管理	赛事管理（UC9）	22	6	0	0
	战队管理（UC12）	40	0	0	0
统计管理	报名统计（UC13）	9	1	0	0
系统参数管理	系统参数配置（UC11）	4	0	0	0
权限管理	登录系统	5	0	0	0
	操作员管理	12	0	0	0
	用户组管理	18	1	0	0
	用户密码修改	2	0	0	0
	Session 同步	4	0	0	0
合 计		245	11	0	8

通过计算可知，该测试版本的测试测试执行通过率为 92.8%。

4、缺陷解决率

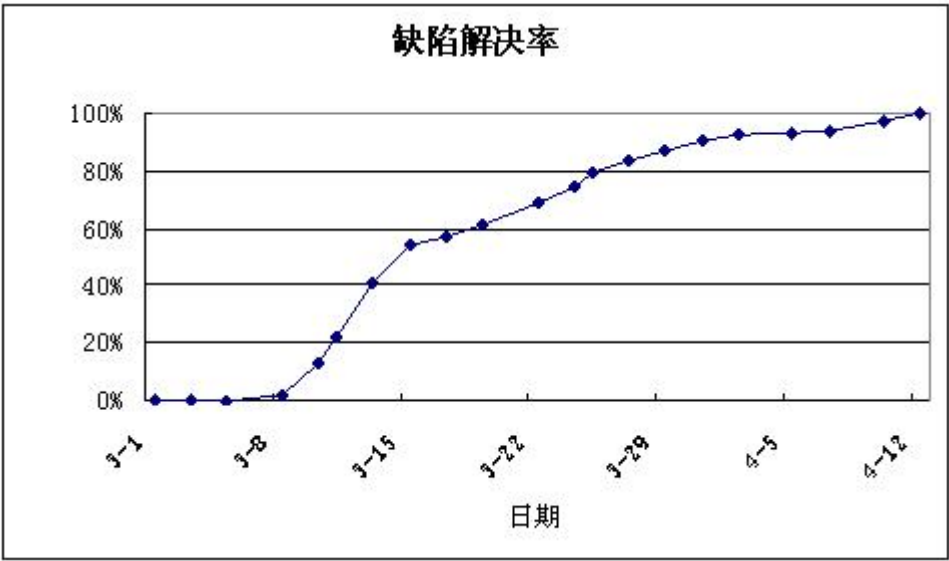
缺陷解决率，指某个阶段已关闭缺陷占缺陷总数的比率。缺陷关闭操作包括以下两种情况：

正常关闭：缺陷已修复，且经过测试人员验证通过；

强制关闭：重复的缺陷；由于外部原因造成的缺陷；暂时不处理的缺陷；无效的缺陷。这类缺陷经过确认后，可以强制关闭。

计算公式：已关闭的缺陷/缺陷总数

在项目过程中，在开始时缺陷解决率上升很缓慢，随着测试 工作的开展，缺陷解决率逐步上升，在版本发布前，缺陷解决率将趋于 100%，如图二所示。一般来说，在每个版本对外发布时，缺陷解决率都应该达到 100%。也就是说，除了已修复的缺陷需要进行验证外，其他需要强制关闭的缺陷必须经过确认，且有对应的应对措施。可以将缺陷解决率作为测试结束和版本发布的一个标准。如果有部分缺陷仍处于打开或已处理状态，那么原则上来说，该版本是不允许发布的。



图二 缺陷解决率

缺陷关闭数据，可以通过缺陷跟踪工具定期（如每周）收集当前系统的缺陷数、已关闭缺陷数，通过这两个数据，即可绘制出整个项目过程或某个阶段的缺陷解决率曲线。

当然了，测试度量指标不仅仅只包括上述四个，在实际工作中，还会用到如：验证不通过率、缺陷密度等指标。收集这些数据目的是为了能对测试过程进行量化管理。但是，简单收集度量数据不是目的，通过对数据的分析、预防问题、对问题采取纠正措施，减少风险才是目的。



李吉岩 (Jiyan Li)
 软件测试咨询顾问
 Office 021-54242775 | Mobile 13636638285 | Fax 021-64471599-8021
 MSN lijian-leo@hotmail.com | QQ 80069119
 上海市徐汇区乐山路 33 号 1 号楼 510 室