

由需求得到回归测试用例排序技术

李华莹, 柴丽雅

(中国电子系统工程研究所, 北京 100141)

摘要: 为了提高回归测试用例集的测试效率和有效性, 提出由需求得到回归测试用例排序技术及其实现算法。由需求得到回归测试用例排序技术, 将与软件需求相关的需求描述度、需求实现复杂度、需求稳定性和需求覆盖度等因素应用于测试用例排序, 以缺陷检测加权平均百分比作为度量标准。通过实验, 比较排序后用例和未排序用例缺陷检测情况, 实验结果表明该技术排序后的回归测试用例集, 能够尽早地发现更多的软件错误, 有效提高回归测试效率, 保证软件质量。

关键词: 软件需求; 测试用例; 用例排序; 回归测试

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2013)11-0070-04

doi: 10.3969/j.issn.1673-629X.2013.11.018

Test Case Prioritization for Regression Testing Based on Requirement

LI Hua-ying, CHAI Li-ya

(Institute of China Electronic System Equipment, Beijing 100141, China)

Abstract: In order to improve the efficiency and validity of regression testing, test case prioritization method based on software requirement for regression testing and prioritization algorithm is proposed in this paper. Four factors related software requirement are taken into account in the method for ranking test cases, the factors are requirement description, requirement implementation complexity, requirement stability and requirement coverage. In this paper, average of the percentage of faults detected is used as a measurement criterion. Comparison of prioritized and non prioritized test case, the experimental results show that, the efficiency of regression testing is improved and the better software quality is achieved by using this method.

Key words: software requirement; test case; prioritization; regression testing

0 引言

在软件进化过程中, 因被测软件的变更, 测试人员需要不断补充测试用例集, 用以验证软件变更的正确性, 这就导致测试用例集的规模不断增加, 测试成本逐渐加大。如果在软件回归测试阶段, 毫无策略地执行测试用例集, 将会耗费约 50% 的软件维护成本^[1-2]。因此, 在软件回归测试阶段, 兼顾测试效率和有效性的同时, 如何执行测试用例, 已成为回归测试的研究热点。

随着软件规模越来越大, 回归测试阶段, 运行所有测试用例常常需要耗时几天甚至几个月; 在此情况下, 测试人员可以按照一定的策略对测试用例进行排序, 使高优先级的测试用例能够较早被执行; 排序后的测试用例集, 优先执行具有最大检错能力的用例, 有效保证软件质量。针对各种不同的应用需求, 人们对测试

用例排序技术进行了大量研究。Kim 等研究了综合考虑各种测试历史的用例排序技术^[3]; Dennis 等研究了基于切片的测试用例排序技术^[4]; 朱海燕等人研究了测试用例排序^[5]; 屈波等人提出了一种基于测试用例设计信息的测试用例优先级方法^[6]。

文中由需求得到回归测试用例排序技术, 将测试用例对应的需求描述度、需求实现复杂度、需求稳定性和需求覆盖度等与软件需求相关的影响因素应用于回归测试用例排序, 有助于尽早发现更多的软件缺陷, 提高回归测试用例执行效率, 节约测试成本。

1 测试用例优先级概念

测试用例优先级是对测试用例按某种特定顺序进行排序, 以最优用例执行顺序覆盖被测软件需求, 更早检测软件缺陷, 提高软件测试效率。测试用例优先

级的研究内容主要是寻找测试用例的排序方法,依据方法确定每个测试用例的优先级,再由测试用例优先级的高低,决定测试用例的执行顺序,以达到最优化测试用例执行效率的目的。

Rothermel^[7-8]定义测试用例优先级概念如下: T 为一组测试用例集, PT 是 T 中测试用例所有可能的排序, f 为从 PT 映射到实数的函数。存在 $T' \in PT$, 对于任何 $T'' \in PT$, 且 $T'' \neq T'$, 存在 $f(T'') \leq f(T')$ 。 f 是对排序目标的定量描述, 用于度量排序的有效性, f 的值越大, 表明测试用例的排序越有效。

2 由需求得到回归测试用例排序

由需求得到回归测试用例排序技术, 从系统测试的角度出发, 结合测试用例的需求描述度、需求实现复杂度、需求稳定性和需求覆盖度等与软件需求相关的影响因素, 融合首次测试时用例对应需求的情况, 设计测试用例集排序并给出测试用例排序算法。

2.1 测试用例排序影响因素

由需求得到回归测试用例排序技术中, 包含四种影响测试用例排序的因素, 这四种因素分别是需求描述度、需求实现复杂度、需求稳定性和需求覆盖度。

2.1.1 需求描述度

需求描述度, 是对软件需求描述的度量。需求描述度主要从描述需求的七个指标进行度量, 这七个指标分别是完整性、正确性、可行性、必要性、优先级、无二义性和可验证性。

需求描述不满足这七个指标中的任一项, 都可能导致不明确的需求, 依据不明确需求编制的软件模块将隐含更多软件缺陷, 其对应的测试用例需要更早被执行; 因此, 在需求描述的七个指标中, 当需求描述越不满足某项指标时, 则该项指标量化值越高, 才能使得该项需求对应的用例具有较高的优先级。

测试用例覆盖的软件需求, 对照需求的七个指标得到指标量化值 d_i ($i=1, 2, \dots, 7$), d_i ($i=1, 2, \dots, 7$) 为 1 到 10 数值区间的值。第 n 个测试用例的需求描述度 RD_n 由下面的公式(1)得到。

$$RD_n = \frac{D_n}{\max(D)} * 10 \quad (1)$$

公式(1)中, $D_n = \sum_{i=1}^7 d_i$ 表示第 n 个测试用例覆盖的软件需求对应七个指标的量化和值; $\max(D)$ 表示测试用例集中单个用例对应需求的七个指标量化和值的最大值。

2.1.2 需求实现复杂度

通常, 越复杂的软件模块具有的软件缺陷越多。软件 80% 的缺陷集中在 20% 的模块中。因此, 考虑将

软件实现复杂度应用于测试用例优先级, 使实现复杂度高的需求对应的用例具有较高优先级, 能够较早被执行, 更早地检测到软件缺陷, 提高测试效率。由下面的公式(2)量化得到第 n 个测试用例的需求实现复杂度 RI_n 。

$$RI_n = \frac{R_n}{\max(R)} * 10 \quad (2)$$

公式(2)中, R_n 表示第 n 个测试用例对应需求的复杂情况, 其值为 1 到 10 的数值; $\max(R)$ 表示单个测试用例对应需求实现复杂度的最大值。

2.1.3 需求稳定度

需求稳定度相对于需求变更而言, 需求变更常常引入软件缺陷, 越频繁的需求变更, 其软件质量越难以保障。如果测试用例对应的需求变更越少, 那么需求稳定度越高, 则用例执行的优先级将越低; 只有这样才能保证变更越频繁的需求, 其对应的用例能被较早执行。由下面的公式(3)得到第 n 个测试用例的需求稳定度量值 RC_n 。

$$RC_n = \frac{C_n}{\max(C)} * 10 \quad (3)$$

公式(3)中, C_n 表示第 n 个测试用例对应需求的变更次数; $\max(C)$ 表示测试用例集中单个测试用例对应需求最大的变更次数。

2.1.4 需求覆盖度

需求覆盖度是对需求能否覆盖软件的关键功能和特性的度量。需求覆盖软件关键功能和特性越多, 其对应用例的执行效率越高, 用例执行优先级也越高。由下面的公式(4)量化得到第 n 个测试用例的需求覆盖度 RS_n 。

$$RS_n = \frac{S_n}{\max(S)} * 10 \quad (4)$$

公式(4)中, S_n 表示第 n 个测试用例对应需求的覆盖情况, 其值为 1 到 10 的数值; $\max(S)$ 表示测试用例集中单个测试用例对应需求覆盖度的最大值。

2.2 测试用例排序

回归测试第 n 个测试用例的优先级取值 RP_n 通过下面的公式(5)得到。

$$RP_n = RD_n + RI_n + RC_n + RS_n \quad (5)$$

软件测试阶段, 根据测试用例优先级影响因素, 由公式(5)得到单个测试用例的优先级, 设计由需求得到回归测试用例的排序算法 TRG。

算法: 用例排序算法 TRG。

输入: 测试用例集 T ;

输出: 排序后的测试用例集 T' 。

Begin

T' 为空;

```
for ( t = 1, t <= | T |, t++)
{ Calculate( RPt ); } //计算每个测试用例的优先级
Sort( T );           //依据测试用例优先级,以降序方式排列测试用例
T' = T;              //将降序排列后的用例集 T 赋给 T'
Output( T' );        //输出测试用例集 T'
End
```

3 度量标准

为了计算排序后的测试用例检测软件缺陷的测试效率,研究者定义了 APFD^[9-11] (average of the percentage of faults detected) 作为度量标准。APFD 的定义详见公式(6)。

$$APFD = 1 - \frac{\sum_{j=1}^k TF_j}{nk} + \frac{1}{2n} \tag{6}$$

式中, n 代表测试用例集 T 的测试用例数目; k 代表被测软件的缺陷数量; TF_j 是检测到缺陷 j 的第一个测试用例在测试用例集 T 中的位置。APFD 的值越大,说明对应测试用例优先级方法检测到缺陷的速率越快。

被测软件 A 中有 10 个软件缺陷,测试用例集包含 10 个测试用例,测试用例缺陷检测情况见表 1。

表 1 测试用例缺陷检测情况表

	测试用例									
	用例 ₁	用例 ₂	用例 ₃	用例 ₄	用例 ₅	用例 ₆	用例 ₇	用例 ₈	用例 ₉	用例 ₁₀
检测到的缺陷	缺陷 F_1	√			√				√	
	缺陷 F_2	√		√	√		√			√
	缺陷 F_3	√						√		
	缺陷 F_4				√	√		√		
	缺陷 F_5			√			√			
	缺陷 F_6	√						√		
	缺陷 F_7						√			√
	缺陷 F_8				√					
	缺陷 F_9			√					√	
	缺陷 F_{10}					√	√			

测试用例执行序列 { $t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}$ }

的 APFD 值为:

$$APFD = 1 - \frac{2 + 1 + 2 + 5 + 1 + 7 + 4 + 5 + 4 + 6}{10 \times 10} + \frac{1}{2 \times 10} = 0.68$$

4 实验分析

为了评价和比较由需求得到回归测试用例排序技

术的有效性,选取聊天软件 P_1 和图片处理软件 P_2 进行测试与分析。针对软件需求,为聊天软件 P_1 设计 125 个系统级测试用例,为图片处理软件 P_2 设计 87 个系统级测试用例,每个软件分别测试两次,第一次测试执行未排序的测试用例,第二次测试执行由需求得到回归测试用例排序技术排序后的测试用例。

聊天软件 P_1 测试用例未排序和排序后的缺陷检测情况如图 1 所示,两次测试用例集的 APFD 值分别为 0.61、0.74。

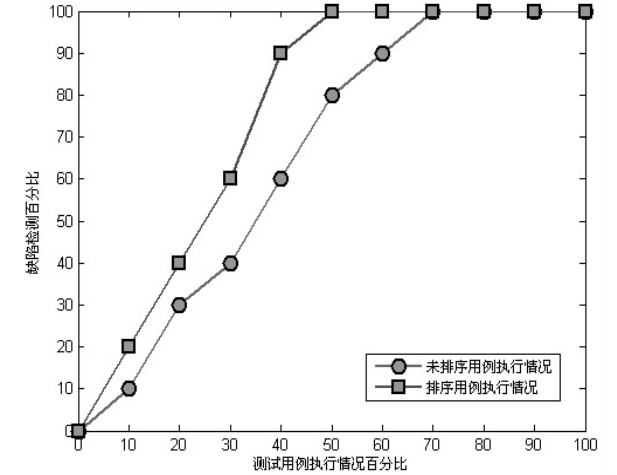


图 1 软件 P_1 测试用例未排序和排序后缺陷检测情况图

图片处理软件 P_2 在测试用例未排序和排序后的缺陷检测情况如图 2 所示,两次测试用例集的 APFD 值分别为 0.54、0.68。

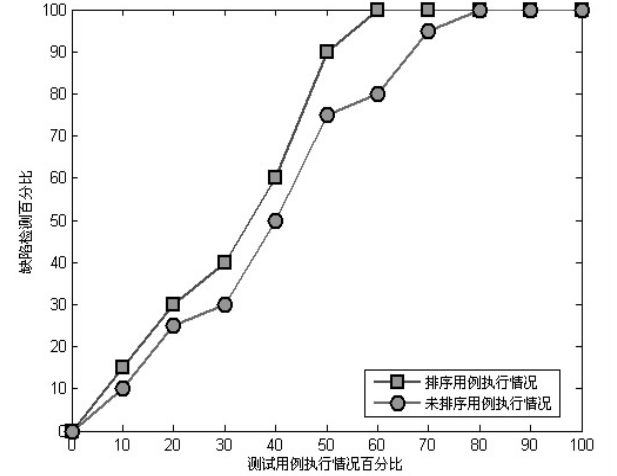


图 2 软件 P_2 测试用例未排序和排序后缺陷检测情况图

实验分析,聊天软件 P_1 和图片处理软件 P_2 分别执行排序后用例集的 40% 和 60% 即可检测出软件的全部缺陷;而对于两个软件未排序的用例集,分别需要执行用例集的 70% 和 80% 的用例才能达到同等测试效果。由此可见,文中提出的排序技术,可以有效提高

回归测试用例集的执行效率。

5 结束语

由需求得到回归测试用例排序技术,通过计算需求描述度、需求实现复杂度、需求稳定度和需求覆盖度四种影响因素对测试用例优先级的影响,得到测试用例优先级取值,按照测试用例优先级取值高低决定用例执行顺序。这种方法有利于尽快检测到程序中的缺陷,提高测试效率;有利于尽快达到测试覆盖率标准,降低测试成本;当测试时间受限时,能够在有限时间内执行更多有效的测试用例,相对充分地保证软件质量。

参考文献:

[1] 李红红,郝克刚,葛 玮. 一个划算的回归测试用例排序的成本-收益模型[J]. 计算机应用与软件,2009,26(4):131-132.

[2] 朱海燕. 软件测试用例集缩减的一个算法[J]. 微电子学与计算机,2007,24(1):204-206.

[3] Kim J M, Porter A. A history-based test prioritization technique for regression testing in resource constrained environments[C]//Proceedings of the 24th international conference on software engineering. New York, USA: ACM, 2002: 119-129.

[4] Dennis J, Neelam G. Test case prioritization using relevant

slices[C]//Proceedings of the 30th annual international computer software and applications conference. Washington D C, USA: IEEE, 2006: 411-420.

[5] 朱海燕,范 辉,谢青松,等. 测试用例排序的研究[J]. 计算机工程与科学,2008,30(1):79-81.

[6] 屈 波,聂长海,徐宝文. 基于测试用例设计信息的回归测试优先级算法[J]. 计算机学报,2008,31(3):431-439.

[7] Ellmn S, Malishevsky A, Rothcmel G. Prioritizing test cases for regression testing[C]//Proceedings of the 2000 ACM SIGSOFT international symposium on software testing and analysis. Portland, Oregon, USA: [s. n.], 2000: 102-112.

[8] Ellmn S, Malishevsky A, Rothcmel G. Test case prioritization: a family of empirical studies[J]. IEEE transactions on software engineering, 2002, 28(2): 159-182.

[9] Ellmn S, Malishevsky A, Rothermel G. Incorporating varying test costs and fault Severities into test case prioritization[C]//Proceedings of the 23rd international conference on software engineering. Toronto, Ontario, Canada: [s. n.], 2001: 329-338.

[10] Rothermel G, Untch R H, Chu C, et al. Prioritizing test cases for regression testing[J]. IEEE transactions on software engineering, 2001, 27(10): 929-948.

[11] 屈 波,聂长海,徐宝文. 回归测试中测试用例优先级技术研究综述[J]. 计算机科学与探索,2009,3(3):225-233.

(上接第 69 页)

参考文献:

[1] 纂宝声. 基于 SQLServer2008 的 DML 触发器设计实例分析[J]. 计算机技术与发展,2012,22(6):229-232.

[2] Agrawal S, Chaudhuri S, Kollar L, et al. Database tuning advisor for Microsoft SQL Server 2005[C]//Proceedings of 2004 VLDB conference. [s. l.]: [s. n.], 2004: 1110-1121.

[3] Natarajan J, Bruchez R, Shaw S, et al. Pro T-SQL 2012 programmer's guide[M]. GRE: Springer, 2012.

[4] 龚 波. SQL Server 2000 教程[M]. 北京:北京希望电子出版社,2005.

[5] 邓卫红,肖卓鹏,黄金水. SQL Server 触发器的妙用[J]. 硅谷,2010,5(24):25-28.

[6] 郑建峰,董国平. 专家门诊-SQL Server 开发答疑 200 问[M]. 北京:人民邮电出版社,2005.

[7] MSDN. DML 触发器[EB/OL]. 2012-12-25. <http://msdn.microsoft.com/zh-cn/library/ms178110.aspx>.

[8] 王军弟,刘瑞玲. SQL Server 中约束与触发器差异比较[J]. 电脑与信息技术,2011,19(2):75-77.

[9] Viera R. Beginning Microsoft SQL Server 2008 programming[M]. USA: Wiley Publishing, 2009.

[10] Microsoft TechNet. 技术资源库[EB/OL]. 2012-12-20. <http://technet.microsoft.com/zh-cn/library/ms186762.aspx>.

[11] Davidson L, Moss J M. Pro-SQL Server 2012 relational database design and implementation[M]. USA: Apress, 2012.

[12] 百度百科. Inserted[EB/OL]. 2012-12-25. <http://baike.baidu.com/view/1473043.htm>.

[13] 梁方明. SQL Server 2000 数据库编程[M]. 北京:北京希望电子出版社,2002.

[14] 黄团利. SQL Server 2005 学习笔记之触发器简介[EB/OL]. 2012-12-15. <http://blog.csdn.net/htl258/article/details/4002984>.

[15] 申玉静,杨会清. SQLServer2005 中利用触发器实现数据库完整性的教学案例设计[J]. 计算机光盘软件与应用, 2012,5(19):240-242.

[16] 胡鹤年. SQL Server 触发器在数据库设计中的应用[J]. 电脑编程技巧与维护,2012,19(8):37-38.

[17] Shapiro J R. Microsoft SQL Server 2005: the complete reference[M]. USA: McGraw-Hill Companies, 2007.

由需求得到回归测试用例排序技术

作者：[李华莹](#)，[柴丽雅](#)，[LI Hua-ying](#)，[CHAI Li-ya](#)
作者单位：[中国电子系统工程研究所, 北京, 100141](#)
刊名：[计算机技术与发展](#)

ISTIC

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2013(11)

本文链接：http://d.wanfangdata.com.cn/Periodical_wjfz201311019.aspx