

基于改进功能点分析法的软件功能测试规模估算模型

陈艳艳 顾勋梅 陈点点

(淮海工学院 计算机工程学院 江苏 连云港 222005)

[摘要] 针对传统功能分析法在复杂度等级划分上的缺陷,提出一个基于模糊功能点分析方法的软件功能测试规模估算模型.在应用功能点分析方法进行测试规模估算时选择适合软件测试领域的规模因子进行调整.最后给出了软件功能测试规模估算流程.

[关键词] 功能测试规模 模糊功能点分析 估算

[中图分类号] TP311 [文献标志码] A [文章编号] 1672-1292(2012)04-0062-05

Software Functional Testing Size Estimation Model Based on an Improved Function Point Analysis

Chen Yanyan ,Gu Xunmei ,Chen Diandian

(School of Computer Engineering ,Huanhai Institute of Technology ,Lianyungang 222005 ,China)

Abstract: Since there are some shortcomings of precision in traditional Function Point Analysis method in the process of dividing complexity grades ,a functional testing size estimation model is produced based on the fuzzy Function Point Analysis method. This model can be available for selecting an appropriate Value Adjustment Factor in order to estimate testing size rightly. Finally ,a detailed flow of estimating testing size is given.

Key words: functional testing size ,fuzzy function point analysis ,estimation

功能点分析法(Function Points Analysis ,FPA)最早由 Albrecht 在 20 世纪 70 年代末提出,其重要意义在于将基于功能特征的度量方法引入软件开发领域.在 30 余年的发展过程中,形成了以 IFPUG 功能点分析法^[1]、COSMIC 全功能点法、Mark II 方法以及 NESMA 功能点法等为代表的超过 20 种不同方法.

功能点分析法在软件度量中已得到广泛应用.该方法基于用户功能性需求(Functional User Requirements ,FUR),以功能点作为度量单位描述软件规模,有以下优点:独立于编程语言和各种技术平台,适合在软件开发需求分析阶段即早期进行项目规模估算,方便与客户基于系统需求的沟通等.正是以上特性,极大地支持了项目管理活动,目前已成为软件项目估算的主流方法^[1,2].

一直以来,人们将软件测试视为是软件开发的一个阶段,因此面向软件测试规模或工作量的估算并未受到重视.然而随着软件外包模式的兴起,测试外包作为软件外包的一种形式已相当普遍.纯测试项目形式、测试系统化和测试人员的专业化也已成为当今软件测试的重要特征.面对新的软件测试形式,进行专门的测试规模估算和度量已成为迫切需求^[3].

在软件项目估算中广泛使用的功能点分析法,并不是专门为软件测试而设计的,但是由于软件功能测试也是面向软件功能的,且 FPA 是从用户需求出发,对于数据文件和元过程的引用在一定程度上体现了功能测试的规模^[3],因此已有研究人员将功能点法引入到了软件测试领域.本文即是在研究功能点法在测试领域应用的基础上,对 IFPUG 法进行改进,用于软件功能测试规模估算.

1 传统 FPA 介绍

IFPUG 功能点法是典型的功能分解类方法,其应用范围已从单纯的规模度量发展到面向整个软件生

命周期.

IFPUG FPA 方法的一般作法是^[1-3]:

- (1) 确定分析类型,分析类型分为 3 种: 新开发项目分析、增强型项目分析、应用程序项目分析.
- (2) 识别项目范围和边界.
- (3) 区分功能类型,功能类型为 2 类: 事务功能类型(EI、EO、EQ) 和数据功能类型(ILF、EIF) .
- (4) 计算未调整功能点 UFP.
- (5) 确定调整因子.
- (6) 计算调整功能点 AFP.

1.1 IFPUG FPA 要素

FPA 要素即为分解的功能类型,共分为 2 大类 5 种: 外部输入(External Input ,EI) ,外部输出(External Output ,EO) ,外部查询(External Query ,EQ) ,内部逻辑文件(Internal Logic File ,ILF) ,外部接口文件(External Interface File ,EIF) ^[1].

1.2 功能要素的复杂度等级

事务功能类型要素的复杂度等级由数据元素类型(Data Element Type ,DET) 和引用文件类型(File Type Referenced ,FTR) 决定,数据功能类型要素的复杂度等级由 DET 和记录元素类型(Record Element Type ,RET) 决定. 复杂度等级确定如表 1 ~ 表 3 所示^[1,4].

表 1 EI 复杂度等级

Table 1 Complexity matrix of EI			
EI	DET		
FTR	1 ~ 4	5 ~ 15	≥16
0 ~ 1	低	低	中
2	低	中	高
3 以上	中	高	高

表 2 EO/EQ 复杂度等级

Table 2 Complexity matrix of EO and EQ			
EO/EQ	DET		
FTR	1 ~ 5	6 ~ 19	≥20
0 ~ 1	低	低	中
2 ~ 3	低	中	高
4 以上	中	高	高

表 3 ILF/EIF 复杂度等级

Table 3 Complexity matrix of ILF and EIF			
ILF/EIF	DET		
RET	1 ~ 19	20 ~ 50	≥51
1	低	低	中
2 ~ 5	低	中	高
6 以上	中	高	高

1.3 复杂度等级与功能点数转换关系

当按上述表 1 ~ 表 3 确定了 5 种要素的复杂度等级后,可按表 4 转换为相应功能点数.

1.4 计算功能点

首先计算未调整功能点(Unadjusted Function Points ,UFP) ^[1],而后根据 14 个通用系统特征(如表 5 所示) 计算调整因子(Value Adjustment Factor ,VAF) :

$$VAF = 0.65 + 0.01 \times \sum_{i=1}^{14} F_i$$

14 个通用系统特征的权重系数取值为 0 ~ 5,分别表示无影响 0 ~ 强相关 5,由此计算出的 VAF 取值范围为 [0.65 ~ 1.35],可对 UFP 产生 ±35% 的调整^[1,4].

最后求出调整后的功能点数(Adjusted Function Points ,AFP) ,AFP = UFP * VAF.

2 改进功能点方法

2.1 传统 FPA 存在的问题

随着软件规模的日益庞大和复杂度的日益增长以及 Web 技术的快速发展,FPA 的缺点也日益显现,主要表现为: 度量过程繁琐,容易发生差错,在 RET 和 GSC(通用系统特征) 的估算上主观性因素较大; 没有考虑各子系统的集成规模; 各功能要素的复杂度等级划分不连续、不精确,尤其对于规模较大的系统估算不准确; 功能点分析方法在实时与科学应用领域方面的运用存在争议^[4-6].

表 4 复杂度等级与未调整功能点对应表

Table 4 Translation table between complexity and unadjusted function points

复杂度	ILF	EIF	EI	EO	EQ
低	7	5	3	4	3
中	10	7	4	5	4
高	15	10	6	7	6

表 5 通用系统特征列表

Table 5 List of general system characteristics

序号	名称	序号	名称
1	数据通讯	8	在线更新
2	分布数据处理	9	复杂处理
3	性能	10	重用性
4	负荷要求	11	安装容易性
5	事务率	12	操作容易性
6	在线数据输入	13	多场所
7	最终用户效率	14	容易更改

对于以上描述的对各类型功能要素的复杂度等级划分不连续问题,结合表 6 详细描述如下:(1) ILF2 和 ILF3 都对应 1 个 RET,但由于 DET 相差 1,而导致其复杂度分别对应低和中,功能点数相差 3。(2) ILF4 和 ILF5 对应 RET 为 6,但其 DET 分别为 20 和 70,相差 50 个 DET,其功能点数却相同,即其功能性相同,这显然与实际不符。表 6 中 ILF1 和 ILF2 也存在相似情况^[4]。

2.2 改进方法

为解决以上问题,采用模糊理论来扩展传统的 FPA,改进方法步骤为:通过产生梯形模糊数来对 FPA 复杂度矩阵中的复杂度等级进行模糊化;产生新的复杂度等级来扩展 FPA 复杂度矩阵,如表 7~表 9 所示;根据新的复杂度等级确定功能点数量,如表 10 所示;消除功能点数值中复杂度等级的模糊性^[4,5,7]。

表 7 扩展后的 EI 复杂度矩阵					表 8 扩展后的 EO/EQ 复杂度矩阵					表 9 扩展后的 ILF/EIF 复杂度矩阵				
Table 7 An extended complexity matrix of EI					Table 8 An extended complexity matrix of EQ/EO					Table 9 An extended complexity matrix of ILF/EIF				
EI		DET			EO/EQ		DET			ILF/EIF		DET		
FTR	1~4	5~15	16~26	≥27	FTR	1~5	6~19	20~33	≥34	RET	1~19	20~50	51~81	≥82
0~1	低	低	中	高	0~1	低	低	中	高	1	低	低	中	高
2	低	中	高	更高	2~3	低	中	高	更高	2~5	低	中	高	更高
3 以上	中	高	高	更高	4 以上	中	高	高	更高	6 以上	中	高	高	更高

针对表 6 应用改进方法得到的功能点数转换结果如表 11 所示。

表 10 改进后的复杂度等级与功能点对应关系						表 11 改进后的 ILF 复杂度及功能点数量转换					
Table 10 Improved translation table between complexity and function points						Table 11 Improved translation table between complexity and function points for ILF					
复杂度	ILF	EIF	EI	EO	EQ		ILF1	ILF2	ILF3	ILF4	ILF5
低	7	5	3	4	3	DET	19	50	51	20	70
中	10	7	4	5	4	RET	1	1	1	6	6
高	15	10	6	7	6	复杂度	低	低	中	高	高
更高	22	14	9	10	9	FP	7	10	10	15	19

由于在复杂度矩阵中,现有的复杂度等级有“低”、“中”和“高”。为使功能点数量更趋精确化,本文增加一个新的复杂度等级“更高”。对于新产生的“更高”等级,其功能点数量是根据每个功能要素现有的复杂度等级进行推算所得到的。本方法采用牛顿插值公式来得到新产生等级的功能点数量,同时引入模糊矩阵隶属度的概念,进行了模糊化消除过程,有效地解决了复杂度等级划分过程中存在的问题^[4]。

3 软件测试规模估算

3.1 规模调整因子

由于功能点分析方法研究公布的目的是为了帮助软件项目解决管理问题,促进项目规范化,其一开始就是面向软件开发或整体项目的,而不是面向软件测试的,而原来功能点分析法中定义的 14 个通用系统特征(GSC)并不完全符合软件测试活动的特点,因此需重新选取调整因子对未调整功能点进行调整,得到面向软件功能测试的功能点估算值^[8-10]。

考虑对功能测试规模产生影响的因素,本文选取需求完备性(CR)、可重用性(RD)、质量级别(QL)和人员因素(HF)。其中,需求完备性(CR)表示需求描述的完备程度、变更的频繁程度以及文档质量。可重用性(RD)表示测试复用程度,取值分别为 0.7(可复用)和 1(不可复用)^[4]。质量级别(QL)表示相应模块(功能)的应用领域复杂度、可靠性和安全性等方面的需求,也间接反映用户对其重要性程度的认

表 12 规模调整因子等级定义			
Table 12 Definition of value adjustment factor grades			
等级	需求完备性 (CR)	质量级别 (QL)	测试人员因素 (HF)
低	3	3	2
中	5	6	4
高	8	10	8

识. 测试人员因素(HF)表示包括测试团队的技能、经验和态度、团队的稳定性、项目团队的关系等.

规模调整因子计算如式(1)所示:

$$DF = ((CR + QL + HF) / 15) * RD. \quad (1)$$

3.2 测试规模

在确定了规模调整因子后,计算调整后的功能点数 AFP:

$$AFP = UFP * DF. \quad (2)$$

测试规模是软件测试项目度量的范畴,鉴于软件测试项目的特点,目前主要采用测试用例数表示测试规模^[10],式(3)~式(6)^[6]可估算测试用例数(Test Case Number,TCN):

$$TCN_{min} = FP^{1.15}. \quad (3)$$

$$TCN_{avg} = FP^{1.2}. \quad (4)$$

$$TCN_{max} = FP^{1.3}. \quad (5)$$

将功能规模 AFP 转换为测试用例数 TCN,以此来描述测试规模,有以下优势:测试用例计数简单、客观,同时可直接反映测试的各个活动;相对稳定的测试团队具有固定的生产效率,可依据 TCN 进一步估算出相应的测试工作量;有利于在测试项目早期开展工作量或成本估算,以合理安排、控制项目进度等.

3.3 软件功能测试规模估算流程

利用改进方法计算功能测试规模的一个实际应用流程可表示如下^[4,10]:

- (1) 把系统划分成 n 个功能单元(模块).
- (2) 分别求出各功能单元(模块)的未调整功能点 UFP_i .
- (3) 分别确定各功能单元(模块)的 DF_i .
- (4) 计算出各功能单元(模块)调整后功能点 AFP_i .
- (5) 求出各功能单元(模块)调整后功能点 AFP_i 之和,以此作为整个系统的 AFP:

$$AFP = \sum_{i=1}^n AFP_i. \quad (6)$$

- (6) 估算测试用例数 TCN.

4 结语

功能点分析方法应用于软件测试领域,用于测试规模的估算是近几年该领域中的研究热点,但由于传统的功能点分析方法并不完全适用于软件测试活动,因此本文在计算未调整功能点数的基础之上选择合理的测试规模因子对其进行调整,使之更能反映软件测试领域的特点.

考虑到传统功能点分析方法 FPA 中复杂度等级划分存在问题,采用了一种基于模糊理论的改进方法进行功能点估算,以更准确计算系统的未调整功能点.

本文最后给出了软件功能测试规模估算流程,并指出在求得整个系统的调整后功能点后,应转换为测试用例数 TCN,以更合理地描述测试规模,并为进一步的工作量(成本)估算做好准备.

[参考文献](References)

- [1] IFPUG. Function Point Counting Practices Manual, Release 4.1 [S]. Mequon, Wisconsin: International Function Point Users Group, 1999.
- [2] Jones C. Applied Software Measurement [M]. 2nd ed. New York: McGraw-Hill, 1996.
- [3] Marín B, Pastor O, Abran A, et al. Towards an accurate functional size measurement procedure for conceptual models in an MDA environment [J]. Data & Knowledge Engineering, 2010, 69(5): 472-490.
- [4] Marín B, Pastor O, Giachetti G, et al. Automating the measurement of functional size of conceptual models in an MDA environment [C]//Proceedings of the Product-Focused Software Process Improvement. Monte Porzio Catone, Italy: Springer, 2008: 215-229.
- [5] Gabriela Robiolo, Cristina Badano, Orosco R, et al. Transactions and paths: two use case based metrics which improve the early effort estimation [C]//3rd International Symposium on Empirical Software Engineering and Measurement. Florida, 2009.

- [6] 顾勋梅,宋国新,邵志清.一种改进的功能点分析方法[J].计算机工程,2007,33(22):12-14.
Gu Xunmei, Song Guoxin, Shao Zhiqing. Improved method function point analysis method[J]. Computer Engineering, 2007, 33(22): 12-14. (in Chinese)
- [7] Ochodek M, Nawrocki J. Enhancing use-case-based effort estimation with transaction types[J]. Foundations of Computing and Decision Sciences, 2010, 35(2): 91-106.
- [8] Ochodek M, Nawrocki J, Kwarciak K, et al. Simplifying effort estimation based on use case points[J]. Information and Software Technology, 2010, 53(3): 200-213.
- [9] 计春雷,闫顺成,宋国新.面向对象方法的 COSMIC-FFP 功能规模度量[J].计算机科学,2011,38(5):119-122.
Ji Chunlei, Yan Shuncheng, Song Guoxin. COSMIC-FFP function size measurement for object-oriented methods[J]. Computer Science, 2011, 38(5): 119-122. (in Chinese)
- [10] 宋巍,沈建雄,杨根兴,等.软件功能测试规模估算模型的研究及其应用[J].计算机工程,2011,37(7):47-49.
Song Wei, Shen Jianxiong, Yang Genxing, et al. Research on size estimation model for software function test and its application [J]. Computer Engineering, 2011, 37(7): 47-49. (in Chinese)

[责任编辑: 严海琳]

(上接第 38 页)

- [10] Ignacio Moreno, Jeffrey A Davis, Travis M Hernandez, et al. Complete polarization control of light from a liquid crystal spatial light modulator[J]. Optics Express, 2012, 20(1): 364-376.
- [11] 李宇波,周强,吴树高,等.液晶型圆偏振调制器[J].光电子激光,2011,22(10):1483-1487.
Li Yubo, Zhou Qiang, Wu Shugao, et al. An optical circular-polarization modulator based liquid crystal[J]. Journal of Optoelectronics•Laser, 2011, 22(10): 1483-1487. (in Chinese)
- [12] 蒋骥,朱宏宇,程柯夫,等.扭曲向列液晶偏振性质的研究[J].实验室研究与探索,2011,30(7):47-49.
Jiang Ji, Zhu Hongyu, Cheng Kefu, et al. Researches on the polarization properties of twisted nematic liquid crystal[J]. Research and Exploration in Laboratory, 2011, 30(7): 47-49. (in Chinese)
- [13] Yeh P, Gu C. Optics of Liquid Crystal Displays[M]. New York: Wiley, 1999.
- [14] 刘永军,宣丽,胡立发,等.高精度纯相位液晶空间光调制器的研究[J].光学学报,2005,25(12):1682-1686.
Liu Yongjun, Xuan Li, Hu Lifa, et al. Investigation on the liquid crystal spatial light modulator with high precision and pure phase[J]. Acta Optica Sinica, 2005, 25(12): 1682-1686. (in Chinese)
- [15] Guo Chengshan, Yue Shujuan, Wang Xilin, et al. Polarization-selective diffractive optical elements with a twisted-nematic liquid-crystal display[J]. Applied Optics, 2010, 49(7): 1069-1074.

[责任编辑: 严海琳]