

目标驱动的软件测试度量方法研究

李兴南^{1,2}, 郝克刚¹, 葛 玮¹

(1. 西北大学 计算机科学系, 陕西 西安 710069;

2. 九江学院 信息科学与技术学院, 江西 九江 332005)

摘 要:文中根据 GSQM 软件度量思想和相关软件度量理论结合软件测试过程的特殊性要求, 提出基于目标驱动的软件测试度量方法(GSTM)。在度量信息需求的基础上主要研究了适合于软件测试度量的基本度量方法、派生度量方法和设计指示器的方法。文中是专门针对软件测试度量进行的研究, 与传统的软件度量理论相比较, 具有很强的针对性、实用性和可操作性。

关键词:软件测试; 软件度量; 基线度量; 派生度量; 指示器

中图分类号: TP311.56

文献标识码: A

文章编号: 1673-629X(2006)09-0103-03

Research on Methods of Goal Drive Software Test Measurement

LI Xing-Nan^{1,2}, HAO Ke-gang¹, GE Wei¹

(1. Department of Computer Science, Northwest University, Xi'an 710069, China;

2. Department of Computer Science and Technology, Jiujiang University, Jiujiang 332005, China)

Abstract: This research in this paper majors in identifying information needs, specifying measures and analysis model of indicator. First put forward the goal-drive software test measurement method named GSTM. Basing the GSTM classed the test measurement information needs into six classes. Basing on the information needs put forward methods to design the base measurement, derived measurement and indicator.

Key words: software test; software measurement; base measurement; derived measurement; indicator

0 引言

为了提高软件的质量, 增加项目的可控制性, 为了定量地评估软件测试的效果, 就必须对软件测试的特性进行度量, 即对影响软件测试的属性进行定量测量^[1]。由于以往的度量理论主要是针对软件开发的整个过程而提出, 单独针对于软件测试过程的度量理论比较少, 同时软件测试是一个相对于软件开发来说比较特殊的过程, 有其独特的属性和要求, 因此文中根据 GSQM^[2] 软件度量思想和其它相关软件度量理论结合软件测试过程的特殊性要求, 提出基于目标驱动的软件测试度量(GSTM, Goal Drive Software Test Measurement)方法。GSTM的主要思路是在进行测试度量时首先采取自顶向下逐步细化的策略分析总结软件测试度量的信息需求, 在确定度量信息需要的基础上设计测试度量的相关方法, 然后实施度量, 在分析研究度量结果的基础上给出度量信息产品供决策者使用并对度量过程进行改进^[3]。

1 测试度量的信息需求研究

根据度量信息模型的关系^[4]可以知道进行度量的目的是为了获得信息产品, 因此 GSTM 度量方法的第一步就是确定测试度量的信息需求, 即确定要解决的问题是什么? 管理者需要什么样的信息产品? 根据软件测试过程的实际情况, 这里将从以下几个方面来分析提取测试度量信息需求:

(1) 为了评估测试过程中采用的测试策略, 比较其优劣、发现缺陷的效率。测试策略一般分为黑盒测试、白盒测试、Web 测试、面向对象的测试。一般将从工作量、成本、效率等几个方面来度量软件测试策略。

(2) 测试过程中由于采用的测试技术不同, 测试的效果也会不一样, 发现缺陷的效率、成本等也会不同。因此要针对各种不同的测试方法设计度量, 比较这些测试方法在发现缺陷的工作量、成本、效率等方面的不同表现。目前, 测试技术主要分为等价类划分、因果图法、正交试验设计法、边值分析法、程序结构分析法、域测试法等。因此必须分别对其设计不同的度量信息需要。

(3) 测试管理和控制方面的目标是为了很好地管理和控制测试过程, 因此要控制测试的进度, 防止进度的延期; 控制测试的成本, 适时地中止测试而不是一味地测试下去, 在成本和软件质量之间寻找一个平衡; 为了完成相关

收稿日期: 2005-12-05

作者简介: 李兴南(1977-), 男, 湖南衡阳人, 硕士研究生, 主要从事软件测试、软件度量研究; 郝克刚, 教授, 博士生导师, 主要从事软件测试、软件度量研究。

的信息需求得出了进度、成本、质量等方面的度量信息需求等。

(4)测试过程的有效性方面,主要是解决测试的缺陷发现率、测试用例和脚本的复用率、回归测试的工作量、测试计划的实施效果等。

(5)测试中间产品方面的度量主要是解决测试用例的设计的效率、平均生产成本、测试用例平均发现缺陷的个数、测试用例的总个数,脚本的设计的效率、平均生产成本、脚本平均发现缺陷的个数、脚本的总个数,缺陷的平均发现成本、工作量等。

(6)测试资源和成本方面主要是解决测试的预算、测试的工作量、测试成本的估算、可以使用的时间、已用的时间等。

2 测试度量的方法研究

在测试度量信息需求确定的基础上为了很好地进行度量必须设计测试度量,在设计度量中要做的主要工作有确定度量策略、构造度量构造、设计基本度量、设计派生度量、设计指示器等。这里主要研究度量技术、度量模型。

2.1 软件测试基本度量

软件测试基本度量:它是指由一个指定的度量方法定义的对软件测试中单个属性的度量,执行该度量将产生一个度量的值。一个软件测试的基本度量在功能上独立于其它所有度量,并获取单个属性的信息。软件测试基本度量的数学描述为 $y = f(x)$,其中 y 为基本度量目标值, f 为度量方法, x 为属性。目前设计基本度量方法主要有简单的统计方法、类推方法、复杂的概率统计参数模型方法^[5,6]等。下面分别对它们进行研究。

a)简单的统计方法 $y = \sum x$,其中 y 为基本度量的目标值, x 为基本度量不可分割的属性。这种方法用于统计代码行数、功能点数、脚本个数,测试用例数、里程碑事件完成数、缺陷个数等。

b)简单的数学关系经验表达式法 $y = ax$,其中 y 为基本度量目标值, x 为基本度量不可分割的属性, a 为经验常数。此方法适用于简单的估算关系,如软件测试用例的设计规模一般可以是 100 行代码设计一个测试用例。

c)复杂的算法模型,比如线性模型 $y = ax + b$,其中 y 为基本度量目标值, x 为基本度量的不可分割的属性, a 为回归常数, b 为回归后的系数。此类方法主要用于比较准确的估算中,比如估算测试工作量、脚本规模、测试用例规模等。表 1 是几个代表性的基本度量。

表 1 基本度量表

基本度量名	度量方法描述和算法	属性描述
估计测试工作量	计算工作总量 $y = a + bx$	x 为被测软件规模
缺陷数统计	计算缺陷总数 $y = \sum x_i$	x 为模块中缺陷数
脚本规模估计	计算脚本行数 $y = ax$	x 为被测软件规模

2.2 软件测试派生度量

软件测试派生度量:软件测试中派生度量是一种基于基本度量的度量,它被定义成两个或多个基本度量或派生度量的一个函数,派生度量获取多于一个属性的信息。软件测试派生度量的数学模型为 $z = g(y_1, y_2, \dots, z_1, z_2, \dots)$,其中 z 为目标派生度量值, g 为度量函数, y_1, y_2, \dots 为基本度量值, z_1, z_2, \dots 为派生度量值。

在 GSTM 方法中根据软件度量的一般理论结合软件测试的实际情况来研究分析测试派生度量。它们主要包括:测试需求方面、测试中间产品方面、测试策略方面等。派生度量主要采用表格的形式来表达,具体形式参见表 2。下面分别从各个派生度量的分类对其进行阐述。

1)测试策略方面,由于在这方面主要是比较各种测试策略在具体测试项目中的优劣,目的多是对他们进行比较,因此大部分采用的度量标度为比率和顺序这两种。派生度量采用的算法有 $y = x_1/x_2$,其中 y 是派生度量目标值, x_1, x_2 是基本度量值,或者是派生度量值。另外还有就是 $y = \text{Max}(x_1, x_2, \dots, x_n)$, $y = \text{Min}(x_1, x_2, \dots, x_n)$ 等,其中 y 为派生度量目标值,取 x_1, x_2, \dots, x_n 的最大值,或者取最小值。表 2 是部分的测试策略方面的派生度量。

表 2 测试策略派生度量表

派生度量名	度量方法描述	度量元或属性的描述
测试策略最大发现缺陷率	$y = \text{Max}(x_1, x_2, \dots, x_n)$	x_1, x_2, \dots, x_n 是白盒或者黑盒测试发现缺陷率
测试策略发现缺陷最低成本	$y = \text{Min}(x_1, x_2, \dots, x_n)$	x_1, x_2, \dots, x_n 是白盒或者黑盒测试缺陷发现成本

2)测试方法方面的度量主要是评价各个度量方法的优劣,这种度量所采用的度量方法主要有 $y = x_1/x_2$,其中 x_1, x_2 是基本度量值或者派生度量值; $y = \text{Max}(x_1, x_2, \dots, x_n)$, $y = \text{Min}(x_1, x_2, \dots, x_n)$,其中 y 为派生度量目标值,取 x_1, x_2, \dots, x_n 的最大值或者最小值。表 3 是部分测试方法方面的派生度量。

表 3 部分测试方法派生度量

派生度量名	度量方法描述	度量元或属性的描述
等价类划分法平均缺陷发现工作量	该法所花费工作量/该法发现的缺陷数	等价类划分法所花费工作量、发现的缺陷数
等价类划分法平均缺陷发现效率	该法发现的缺陷数/该法人时总数	等价类划分法发现的缺陷数、花费的人月总数
正交试验设计法平均缺陷发现成本	该法所花费总成本/该法发现的缺陷数	正交试验设计法总成本、该法发现的缺陷数

3)测试中间产品的派生度量,主要度量中间产品的规模、状态、分布等属性。用到的数学模型是分类模型和顺序模型。比如测试用例的规模、脚本的规模、缺陷的发现数、缺陷的发现成本等。

4)回归测试阶段的派生度量,主要度量回归测试的效率、测试用例和测试脚本的重用率、边际效益、测试成本和工作量趋势等。在这里主要的数学模型是边际效益模型:

$Y = \frac{\Delta x_1}{\Delta x_2}$,其中 $\Delta x_1, \Delta x_2$ 是在回归测试中某属性的增量;

趋势函数 $Y = \lim_{m \rightarrow \infty} \frac{\frac{1}{n} \sum_{j=1}^m X_{ij}}{\frac{1}{m} \sum_{j=1}^m \frac{1}{n} \sum_{i=1}^n X_{ij}}$, 其中 X_{ij} 为第 j 次回归

测试第 i 个属性值, Y 为目标值。例如, 要度量第 j 次回归测试的测试用例生产工作量趋势, 首先计算第 j 次回归测试的平均测试用例生产工作量 $\frac{1}{n} \sum_{i=1}^n X_{ij}$, 然后计算前 j 次测试用例的平均生产工作量 $\frac{1}{j} \sum_{k=1}^j \frac{1}{n} \sum_{i=1}^n X_{ik}$, 再将它们取比值的极限得到第 j 次回归测试的测试用例工作量趋势值 Y , 这是一个与单位无关的绝对值, 如果 Y 的极限是零则说明回归测试阶段的测试用例的复用率效率很高, 几乎为 100% 的复用。表 4 是部分回归测试阶段的派生度量。

表 4 部分回归测试派生度量

派生度量名	度量方法描述	度量元或属性的描述
脚本边际生产率	Δ 脚本数/ Δ 人月	脚本成本, 人月
脚本生产边际工作量	Δ 工作量/ Δ 脚本数	工作量, 脚本数
测试用例生产工作量趋势	第 1 次回归测试用例生产工作量/平均用例生产工作量	第 1 次回归测试用例生产工作量、平均用例生产工作量

5) 测试进度进展、过程控制方面的度量主要解决测试过程的控制方面。主要有比率、顺序类型的度量方法。表 5 列出了部分该方面的派生度量。

表 5 部分测试进度进展派生度量

派生度量名	度量方法描述	度量元或属性的描述
测试进度偏差	实际时间/计划时间	日历表时间, 计划时间
测试成本偏差	实际成本/计划成本	实际成本, 计划成本
测试工作量偏差	实际工作量/计划工作量	实际工作量, 计划工作量

2.3 软件测试度量指示器

软件测试度量指示器, 它是提供指定属性的估计或评价的度量, 该属性是从涉及已定义的信息需要的分析模型中派生。指示器一般包括度量元、决策准则, 其形式化描述如下:

$$\begin{cases} W = h(y_1, y_2, \dots, z_1, z_2, \dots) \\ w \in g, \text{其中 } g \text{ 为决策准则或决策条件} \end{cases}$$

其中 W 为指示器目标值, h 为度量分析模型, y_1, y_2, \dots 为基本度量值, z_1, z_2, \dots 为派生度量值。

在测试度量设计指示器时根据度量信息需求首先要设计基本度量元、派生度量元, 然后设计和选择分析模型, 最后配置决策准则来完成指示器的设计。在实施指示器时通过计算 $h(y_1, y_2, \dots, z_1, z_2, \dots)$ 来得出 W 的值, 然后根

据决策准则即参照系统来分析度量结果并为决策提供信息。指示器的分析模型主要包括比率模型、顺序模型、回归模型等。本研究中对于指示器的表达方式采取表格的形式, 表 6 是部分指示器。

表 6 部分指示器表

指示器名	分析模型	决策准则描述	属性描述
用例设计有效性	缺陷数/用例成本	是否大于决策阈值	缺陷数, 成本
用例复用率	用例边际工作量/用例平均工作量	是否大于决策阈值	用例边际工作量、用例平均工作量
中止测试脚本生产成本临界值	脚本边际成本的极限	是否小于决策阈值	脚本边际成本
中止测试脚本生产工作量临界值	脚本边际工作量的极限	是否小于决策阈值	脚本边际工作量

3 结束语

根据测试过程的特殊需求属性研究了测试度量的信息需求, 通过度量信息需求来驱动度量, 在度量信息需求的基础上研究了在基本度量、派生度量的度量函数方法和设计指示器的方法。其中基于目标驱动的软件度量研究、回归测试的度量研究等方面具有一定的开创性, 当然这一切研究工作还有待在后续研究工作中不断深入下去。

参考文献:

[1] Fenton N. Software measure: a necessary scientific basis[J]. IEEE Trans on Software Engineer, 1994, 20(3): 199-206.

[2] Park R E, Goethert W B, Florac W A. Goal-Driven Software Measurement[Z]. CMU/SEI-96-HB-002. USA: Software Technology Institute of Carnegie Mellon University, 1996. 19-23.

[3] Park R E. Software Size Measurement: A Framework for Counting Source Statements[Z]. CMU/SEI-92-TR-020. USA: Software Technology Institute of Carnegie Mellon University, 1996. 19-24.

[4] McGarry J, Card D, Jones C. Practical Software Measurement Objective Information for Decision Makers[M]. Boston, MA: Pearson Education Inc., 2003. 32-33.

[5] Orr G, Reeves T E. Function point counting: one program's experience[J]. The Journal of Systems and Software, 2000, 53: 239-244.

[6] 温鹏程. 统计分析在软件测试中的应用[J]. 武汉理工大学学报, 2004, 12(5): 22-27.

(上接第 102 页)

参考文献:

[1] Pitoura E, Bhargava B. Data Consistency in Intermittently Connected Distributed Systems[J]. IEEE Transactions on Knowledge and Data Engineering, 1999, 11(6): 896-915.

[2] 张洁. 移动数据库系统 SwifDB 同步机制的设计和实现[D]. 南京: 东南大学, 2003.

[3] Yeo L H, Zaslavsky A. Submission of transactions from mobile

workstation in a cooperative multidatabase processing environment[A]. The 14th Int'l Conf on Distributed Computing Systems[C]. Poland: [s.n.], 1994.

[4] 徐立臻, 江明菲, 董逸生. 移动计算环境下的数据库事务管理[J]. 东南大学学报, 2002, 32(6): 886-889.

[5] 王能斌. 数据库系统教程[M]. 北京: 电子工业出版社, 2002.