



中华人民共和国国家标准

GB/T 29834.3—2013

系统与软件维护性 第3部分：测试方法

Maintainability of system and software—
Part 3: Testing method

2013-11-12 发布

2014-02-01 实施

中华人民共和国国家质量监督检验检疫总局 发布
中国国家标准化管理委员会

目 次

前言 III

引言 IV

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 概述 1

5 易分析性 2

5.1 失效诊断的效率 2

5.1.1 失效诊断的准确性 2

5.1.2 失效诊断的时间 2

5.2 对失效诊断的支持 3

5.2.1 有效线索比例 3

5.2.2 可理解线索比例 4

5.2.3 审核追踪能力 4

6 模块化 5

6.1 模块间的耦合性 5

6.2 模块结构合理性 6

7 规范性 7

7.1 代码易读性 7

7.1.1 注释的充分性 7

7.1.2 注释的规范性 7

7.1.3 代码的规范性 8

7.1.4 代码规范的符合性 9

7.2 文档维护指导性 9

7.2.1 对维护的指导性 9

7.2.2 文档与软件的适应程度 9

7.3 数据的规范性 10

7.3.1 数据的规范性 10

8 易改变性 11

8.1 可修改性 11

8.1.1 代码的可修改性 11

8.1.2 可配置性 11

8.2 修改实施的效率 12

8.2.1 变更周期的效率 12

8.2.2 修改实施的效率 12

8.2.3 修改的复杂度 13

8.3 修改的可控制性..... 13

 8.3.1 修改的可还原性..... 13

 8.3.2 软件变更控制的能力..... 14

9 稳定性..... 14

 9.1 变更成功的比率..... 14

 9.2 修改影响的局部化..... 15

10 可验证性 15

 10.1 可自动验证性 15

 10.2 测试的重启性 16

 10.3 维护完整性 16

附录 A（资料性附录） 代码分析示例 17

参考文献 19

前 言

GB/T 29834 在《系统与软件维护性》总标题下,分为如下 3 部分:

- 第 1 部分:指标体系;
- 第 2 部分:度量方法;
- 第 3 部分:测试方法。

本部分为 GB/T 29834 的第 3 部分。

本部分按照 GB/T 1.1—2009 给出的规则起草。

本部分由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本部分起草单位:上海计算机软件技术开发中心、中国电子技术标准化研究院、深圳市中联信信息技术有限公司、北京邮电大学、上海浦东软件平台有限公司、上海宝信软件股份有限公司、上海鲁齐信息科技有限公司、辽宁北方实验室有限公司、南宁市平方软件新技术有限责任公司、上海市电力公司信息中心。

本部分主要起草人:蔡立志、刘振宇、李晓庆、黄毅、熊琬、张建良、潘国瑞、张旻旻、袁玉宇、李家宏、张露莹、崔岩、丁志刚、徐旻之、毛瑞、吴玉军、胡芸、陈达丽、陈强。

引 言

由于交付的软件存在缺陷、用户需求的变更、环境的变化,软件常常需要进行维护,GB/T 8566—2007 指出软件维护已经成为软件生命周期的一个重要组成部分。GB/T 16260.1—2006《软件工程产品质量 第1部分:质量模型》包含了软件维护性的陈述。GB/T 29834 的本部分参照 GB/T 16260.1—2006 的维护性陈述提出了维护性测试方法。开发方、需方、质量管理者和第三方评测机构可根据指标体系定义维护性质量需求、评价软件产品维护性、测量维护性质量情况或作其他用途。本部分适用于有维护需求的各类系统与软件,不适用于“一次性使用”或者在使用期间无需变更的系统与软件。

本标准预期的主要使用者包括:

a) 软件供方,当:

- 1) 需要声明软件产品维护特性时;
- 2) 对照声明的维护性特性自行评估其系统与软件时;
- 3) 对软件进行维护性相关的产品设计和实现时;
- 4) 为维护性符合性证书或标志进行测试时的第三方评测机构;

b) 潜在的需方,当:

- 1) 对即将采购的软件产品的维护性要求和现有产品的说明信息进行比较时;
- 2) 需要对产品的性能做进一步的改进或者完善、产品的潜在差错而作必需的更改,实际运行的环境和采购的环境存在差异时;
- 3) 检验维护性要求是否被满足时。

GB/T 29834.1《系统与软件维护性 第1部分:指标体系》给出了维护性度量指标体系,GB/T 29834.2《系统与软件维护性 第2部分:度量方法》给出了如何获得维护性指标测量值的度量方法。本部分旨在和 GB/T 29834.1 和 GB/T 29834.2 联合使用。

系统与软件维护性

第 3 部分：测试方法

1 范围

GB/T 29834 的本部分规定了如何获得维护性指标测量值的测试方法。

本部分适用于有维护需求的各类系统与软件。本部分不适用于“一次性使用”或者在使用期间无需变更的系统与软件。

注：本部分中所指的系统主要是软件系统。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 11457 软件工程术语

GB/T 29834.1 系统与软件维护性 第 1 部分：指标体系

GB/T 29834.2 系统与软件维护性 第 2 部分：度量方法

3 术语和定义

GB/T 11457 和 GB/T 29834.1 中界定的术语和定义适用于本文件。

4 概述

GB/T 29834.2 规定了系统与软件维护性的度量方法，本部分针对度量方法中的相关公式，给出测试方法。常见的维护性测试方法有：专家评审法、技术测试法和用户调查法。各类方法说明如下：

- a) 专家评审法——一种主观的测评方法。评审时，应根据被评审对象和评审目的，设计评审项目表，列出打分栏目、分值、权重和打分规则。可由 N 个专家组成一个评审组，专家根据自身的经验与认知，进行判断打分。然后，依据专家的权重和统计规则，进行分值汇总计算，其计算得出的值作为评审的结果。打分规则可以是分等级的，也可以是一个取值区间，或选择一个其他的合适规则；
- b) 技术测试法——一种客观的测评方法。技术测试时，可依据被测对象和测试目的，选择采用适用的自动化测试工具进行，也可由人工进行手工测试。技术测试获得的结果通常是一种量化的测量结果；
- c) 用户调查法——一种面向特定用户群的问卷征询方法。用户调查时，应根据调查的目的和特定的用户群，设计调查表，让被调查对象填写并反馈，调查表的回收数应达到一定的数量，并不低于发出数的适当比例。然后，对回收的调查表进行汇总计算，其计算得出的值作为用户调查的结果。

5 易分析性

5.1 失效诊断的效率

5.1.1 失效诊断的准确性

失效诊断的准确性测试要求如下：

——前置条件

维护者已经实施了维护,并形成了有效的维护记录；

——测试输入

需求规格说明书、设计规格说明书、维护需求、维护计划、维护记录；

——测量类型

计数/计数；

——测试方法

专家评审法；

——实施要点：

- 1) 确定失效诊断效率的测试输入,包括需求规格说明书、设计规格说明书、维护需求、维护计划、维护记录；
- 2) 根据系统与软件的维护相关文档,评审确定系统与软件实际需要进行维护的失效(维护点)个数,并计数为 B ；

注：最直接的、最常见的失效个数获取来源为各种缺陷追踪系统。维护的线索,特别是失效的界面提示信息的截图也通常包含在失效的描述中。

- 3) 针对每一个维护记录,收集相关的维护线索,维护线索宜包括：

- 系统日志(例如操作系统、应用服务器)；
- 操作日志(包括操作者、时间、位置、关键的交易数据、及其有关场景、出错信息)；
- 缺陷界面提示信息；

- 4) 评审维护线索是否直接给出失效分析原因并能定位失效点。直接的失效定位可包括但不限于：

- 维护线索给出错误所在的代码,并说明了原因；
- 维护线索给出了直接的错误原因,如主键冲突、磁盘写操作失效；

示例 1：应用程序在数据增加或者更新时,遇到主键冲突而导致失效,系统界面提示类似“Duplicate entry '1' for key 'PRIMARY'”的失效提示,维护者依据该信息可以直接定位失效的原因。

示例 2：在应用保存数据时,由于存储空间不足而导致磁盘写操作失效,在操作日志中提示类似“Not enough space for environment”的操作日志,维护者可以直接定位失效的原因。

- 5) 统计维护者依据维护线索而无需借助于其他复杂方法能成功分析失效原因的维护点个数,将其计数为 A ；
- 6) 根据 GB/T 29834.2 计算公式 $X=A/B$,计算出测量结果。

5.1.2 失效诊断的时间

失效诊断的时间测试要求如下：

——前置条件

维护者已经实施了维护,并形成了有效的维护记录；

——测试输入

需求规格说明书、设计规格说明书、维护需求、维护计划、维护记录；

- 测量类型
计时/计数；
- 测试方法
技术测试法,对失效诊断的过程进行计时；
- 实施要点：
 - 1) 确定失效定位时间的测试输入,包括需求规格说明书、设计规格说明书、维护需求、维护计划、维护记录；
 - 2) 统计所有维护的维护点数,计数为 n ；
 - 3) 对于每一个维护点,利用维护记录记录维护耗费的时间；
注：适当时,维护记录可包括维护者的能力水平、维护者从业时间。
 - 4) 在维护者完成维护结束时,对每个维护点,记录维护者成功定位失效所耗费的时间,计数为 T_i 。该时间不应包括维护过程中暂停、挂起等其他因素耗费的时间。
 - 5) 根据 GB/T 29834.2 计算公式 $X = (\sum_{i=1}^n T_i)/n$,计算出测量结果。

5.2 对失效诊断的支持

5.2.1 有效线索比例

有效线索比例测试要求如下：

- 前置条件
维护者已经实施了维护,并形成了有效的维护记录；
- 测试输入
需求规格说明书、设计规格说明书、维护需求、维护计划、维护记录等；
- 测量类型
计数/计数；
- 测试方法
专家评审法,对收集的线索按照是否有效进行投票表决；
- 实施要点：
 - 1) 依据需求规格说明书、设计规格说明书、维护需求、维护计划、维护记录等,确定失效诊断支持的测试输入；
 - 2) 在执行维护之前,先备份并初始化系统线索,记录初始的线索数。维护线索宜包括：
 - 系统日志(例如操作系统、应用服务器)；
 - 操作日志(包括操作者、时间、位置、关键的交易数据及其有关场景、出错信息)；
 - 缺陷界面提示信息；注：密切关联的、作为一个整体记录的多个维护记录字段,计数为一条线索。
 - 3) 根据系统与软件的维护相关文档,运行缺陷重现的步骤,在结束时记录并确定系统与软件的计划维护(及相关)点的线索数；
 - 4) 将运行终止时线索数减去初始线索数,计为 B ；
 - 5) 根据维护依据,采用专家评审法评审收集的线索,对每一个线索是否有效进行表决。在评审线索有效性时,专家的组成可包括维护人员、开发人员、测试人员等。统计系统与软件实际能够提供有效线索的个数,并计为 A ；
注 1：带指导意义的错误提示信息,可计为一个有效线索。
注 2：线索的有效性指该线索有效改善了故障的定位时间,或确定了需维护的物理位置。
注 3：如进行不同的系统与软件进行比对,评审的专家组成宜一致。

GB/T 29834.3—2013

6) 根据 GB/T 29834.2 计算公式 $X=A/B$, 计算出测量结果。

5.2.2 可理解线索比例

可理解线索比例测试要求如下:

——前置条件

维护者已经实施了维护,并形成了有效的维护记录;

——测试输入

需求规格说明书、设计规格说明书、维护需求、维护计划、维护记录等;

——测量类型

计数/计数;

——测试方法

专家评审法、对收集的有效线索按照是否可理解进行投票表决;

——实施要点:

1) 确定实施分析能力的测试输入,包括需求规格说明书、设计规格说明书、维护需求、维护计划、维护记录等;

2) 在执行维护之前,先备份并初始化系统线索,记录初始的线索数。维护线索宜包括:

- 系统日志(例如操作系统、应用服务器);
- 操作日志(包括操作者、时间、位置、关键的交易数据及其有关场景、出错信息);
- 缺陷界面提示信息;

注:密切关联的、作为一个整体记录的多个维护记录字段,计数为一条线索。

3) 根据系统与软件的维护相关文档,运行缺陷重现的步骤,在结束时记录并确定系统与软件的计划维护(及相关)点的线索数;

4) 将运行终止时线索数减去初始线索数,计为 B ;

5) 对于所有有效维护线索,采用专家评审法评审收集的线索,对每一个线索是否可理解进行投票表决。在评审线索可理解性时,专家的组成可包括维护人员、开发人员、测试人员等。在可理解性评审和有效性评审,可在一次评审中同时进行,统计系统与软件实际能够提供可理解的线索个数,并计为 A ;

注 1: 采用不同于软件的原始开发者的维护者进行度量该指标。

注 2: 有助于正确理解的线索要素可能包括故障出现的上下文环境、用户权限、当前操作、变量状态、故障提示、维护的建议等。

6) 根据 GB/T 29834.2 度量公式 $X=A/B$, 计算出测量结果。

5.2.3 审核追踪能力

审核追踪能力测试要求如下:

——前置条件

维护者已经实施了维护,并形成了有效的维护记录;

——测试输入

需求规格说明书、设计规格说明书、维护需求、维护计划、维护记录等;

——测量类型

计数/计数;

——测试方法

评审法;

——实施要点:

- 1) 确定实施分析能力的测试输入,包括需求规格说明书、设计规格说明书、维护需求、维护计划、维护记录等;
- 2) 审阅维护相关文档,确定计划在运行中要记录的足以监视系统与软件状态的数据数,并计数为 B ;
注:在维护文档中,计划要监视的数据宜指明数据格式、字段、数据计数的单位。
- 3) 备份并初始化系统状态记录数,记录初始的记录数据数;
- 4) 根据维护文档和需求文档,执行和状态记录相关的功能;
- 5) 运行结束时,统计系统实际记录的状态相关的数据数;
- 6) 根据终止记录数据数和初始记录数据数,计算本次运行记录到的数据数,并计数为 A ;
- 7) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

6 模块化

6.1 模块间的耦合性

模块间的耦合性测试要求如下:

- 前置条件
系统设计已经完成,模块之间的关联关系已经确定;
- 测试输入
系统设计说明书;
- 测量类型
计数/计数;
- 测试方法
技术测试法、专家评审法。宜采用代码静态扫描的技术测试法或者专家评审法,确定模块间的耦合性;
- 实施要点:
 - 1) 查看软件的设计或需求文档,在度量之前确定模块的粒度(如子系统、构件、包、类)等。
 - 2) 统计软件的模块数 n ,对软件的模块从 1 到 n 进行编号;
 - 3) 确定模块间关联强度 C_{ij} 的计算方法,在对类似的软件进行横向评价时,应选用同样的计算方法;
 - 4) 使用已经确定的 C_{ij} 的计算方法,对各个模块间的关联强度进行计算;
注 1: 模块间的单向接口调用不能理解为模块之间的关联。
注 2: 不同模块之间存在共享的全局变量或者其他破坏模块局部性的联系计为关联。
注 3: 不同模块之间的存在循环闭环调用可计为关联。
 - 5) 对每个功能模块,计算 B_i 数值;
 - 6) 根据 GB/T 29834.2 度量公式 $X = \sum_{i=1}^n B_i/n$, $B_i = (\sum_{j=1}^n C_{ij(i \neq j)})/(n-1)$,计算出测量结果。
示例:
目标软件包含有 5 个模块。
步骤:
 - (1) 查看相关的文档,确定该软件有 5 个模块,确定 $n=5$;
 - (2) 确定 C_{ij} 的计算方法为:当模块 i 和模块 j 之间存在关联关系时, $C_{ij}=C_{ji}=1$,否则, $C_{ij}=C_{ji}=0$;

(3) 确定每个 C_{ij} 的数值,可使用矩阵,在本例中,模块 2 和模块 3、模块 3 和模块 4 之间存在关联关系:

C_{ij}	M_1	M_2	M_3	M_4	M_5
M_1	0	0	0	0	0
M_2	0	0	1	0	0
M_3	0	1	0	1	0
M_4	0	0	1	0	0
M_5	0	0	0	0	0

(4) 计算 B_i 的数值: $B_1=0, B_2=0.25, B_3=0.5, B_4=0.25, B_5=0$;

(5) 计算 $X=1/5=0.2$ 。

6.2 模块结构合理性

模块结构合理性测试要求如下:

- 前置条件
系统模块代码编写已经完成;
- 测试输入
系统模块代码、预定义的代码规范;
- 测量类型
计数/计数;
- 测试方法
技术测试法、专家评审法。宜采用代码静态扫描的技术测试或者专家评审,确定模块结构的合理性;
- 实施要点:
 - 1) 确定实施分析能力的测试输入,包括系统模块代码、预定义的代码规范;
 - 2) 根据系统与软件的维护相关文档(如维护需求文档),在度量之前确定模块的粒度(如子系统、构件、包、类)等;
 - 3) 依据粒度的大小,统计系统与软件的总模块数,并计数为 B ;
 - 4) 定义的模块结构的规范,一般情况下,宜检查所有的模块化规范性。也可根据项目的实际情况,定义模块化通过的准则,如违反规则的规则条数占总规则条数的比例应少于一定的阈值(如少于 10%)。
 - 5) 采用静态扫描的方法,根据预定义的规范对指定的模块进行扫描,也可采用专家评审法对指定的模块进行规范化评审。评审内容可包括但不限于以下几个方面:
 - GB/T 14079 列举的常见的不规范模块结构例子包括:存在多个入口和多个出口的模块、执行相同或者相似功能的模块、过多使用嵌入的常量或者字符串等;
 - 可以采用接口而使用其他实现方法而导致的紧耦合、每个模块(类)的代码行大小等;
 - 面向对象代码的规范性,包括了继承数的深度 DIT(Depth of inheritance tree)、子女的数量 NOC(Number of Children)、对类的响应 RFC(Response for a class)等;
 - 6) 统计符合模块化要求的模块数并计数为 A ;
 - 7) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

7 规范性

7.1 代码易读性

7.1.1 注释的充分性

注释的充分性测试要求如下：

- 前置条件
系统代码编写已经完成,并已经通过冒烟测试;
- 测试输入
系统代码;
- 测量类型
计数/计数;
- 测试方法
技术测试法、专家评审法。一般采用利用代码静态扫描的方法对代码注释的充分性进行扫描、并辅以适当的专家评审;
- 实施要点:
 - 1) 一般情况下,宜对所有代码进行注释充分性检查。在无法满足的情况下,查看设计文档(如需求文档、设计文档等),确定软件的选取模块,选取模块宜遵循如下原则:
 - 核心业务代码;
 - 使用频度高的代码;
 - 文档规定的代码;
 - 高优先级的代码;
 - 2) 确定充分性计算的代码单位粒度,宜采用方法或者代码行作为单位;
 - 3) 查看选取模块,确定选取模块中代码单位,计为 B ;
 - 4) 利用静态代码扫描工具对源代码进行代码充分性分析,或采用专家评审(人员阅读),确定选取模块中已加注释的代码单位个数,计为 A ;
 - 5) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。一个代码易读性样例参见附录 A。

7.1.2 注释的规范性

注释的规范性测试要求如下：

- 前置条件
系统代码编写已经完成,并已经通过冒烟测试;
- 测试输入
预定义的注释规范、系统代码;
- 测量类型
计数/计数;
- 测试方法
技术测试法、专家评审法。一般采用利用代码静态扫描的方法对注释的充分性进行扫描、并辅以适当的专家评审;
- 实施要点:
 - 1) 查看软件编码规范文档,确定软件编写时应遵守的注释规范;

- 2) 一般情况下,宜对所有代码进行注释的规范性检查。若无法实现所有代码的检查,宜采用选取方法选取要检查的代码,其选取数量为 B ,选取代码模块宜遵循如下原则:
 - 核心业务代码;
 - 使用频度高的代码;
 - 文档规定的代码;
 - 高优先级的代码。
- 3) 利用静态代码扫描工具检查所选取的注释,也可采用人工评审的方法检查所选取的注释。规范性的要求宜包括但不限于以下几个方面:
 - 常见的注释规范包括:注释位置、注释的内容要求等;
 - 类定义的注释信息,包括:作者;日期;模块功能描述(如功能、主要算法、内部各部分之间的关系、对方法返回值的说明、对方法中某参数的说明、对方法可能抛出的异常进行说明、相关主题等);
 - 文件注释通常要求包括该文件与其他文件关系、本文件历史修改记录;版本信息等;
- 4) 统计符合规范且易于维护人员理解的注释数量,计为 A ;
- 5) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

7.1.3 代码的规范性

代码的规范性测试要求如下:

——前置条件

系统代码编写已经完成,并已经通过冒烟测试;

——测试输入

预定义的代码规范、系统代码;

——测量类型

计数/计数;

——测试方法

技术测试法、专家评审法。一般采用利用代码静态扫描的方法对注释的充分性进行扫描、并辅以适当的专家评审;

——实施要点:

- 1) 查看软件编码规范文档,确定代码编写规范;
- 2) 确定代码规范性的单位粒度,可采用方法或者代码行作为单位;
- 3) 一般情况下,宜对所有代码进行规范性检查。若无法实现所有代码的检查,选取要检查的代码,选取代码模块宜遵循如下原则:
 - 核心业务代码;
 - 使用频度高的代码;
 - 文档规定的代码;
 - 高优先级的代码;
- 4) 统计所有代码或者按照上述原则选取的代码行数,计为 B ;
- 5) 利用静态代码检查工具或者利用专家评审法检查所选取的代码,符合规范的程度。一般的规范可包括但不限于以下几个方面:
 - 常见的规范,包括:变量、过程、函数、参数、类、包等的命名规范;
 - 文件规范,包括:目录的命名规范、文件的命名规范;
 - 排版规范,包括:关键词、变量、语句、函数、异常、块等方面的排版规范;
 - 嵌套(内嵌类、或者递归)定义的深度过大等;

- 6) 统计符合规范要求的代码的行数,计为 A ;
- 7) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

7.1.4 代码规范的符合性

规范的符合性测试要求如下:

- 前置条件
编码规范已定义,代码编写已经完成;
- 测试输入
预定义的编码规范文档、被测源代码;
- 测量类型
计数/计数;
- 测试方法
技术测试法。选择采用代码静态扫描检查代码符合的规范个数;
- 实施要点:
 - 1) 依据软件编码规范文档,选择应遵守的代码编写规范。选择的原则包括:
 - 安全性规范一般宜作为要遵守的规范;
 - 依据历史经验,预计会大量使用的规范;
 - 相关文档规定的特定规范;
 - 2) 统计所有要遵循的规范个数,计为 B ;
 - 3) 对所提交测试的代码,进行检查,获得遵守的规范个数,计为 A ;
 - 4) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

7.2 文档维护指导性

7.2.1 对维护的指导性

对维护的指导性测试要求如下:

- 前置条件
维护者已经实施了维护,并形成了有效的维护记录;
- 测试输入
维护文档、维护记录、系统维护调查表;
- 测量类型:
计数/计数;
- 测试方法
用户调查法;
- 实施要点:
 - 1) 统计维护表中的问题个数,计为 B ;
 - 2) 根据软件维护计划,维护人员实施维护活动,记录实际通过文档能够解决的问题。
注 1: 文档对于维护的指导通常以“维护手册”、“常见问题”或者“已知缺陷”等形式出现。
注 2: 系统自身提供维护功能(数据字典、权限设置、流程定义)解决的问题,也应计算之内。
注 3: 自动更新的系统与软件,文档已经指明更新服务器的地址和端口。
 - 3) 在维护工作结束时,统计借助文档帮助解决的问题个数,计为 A ;
 - 4) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

7.2.2 文档与软件的适应程度

文档与软件的适应程度测试要求如下:

- 前置条件
维护者已经实施了维护,并形成了有效的维护记录;
- 测试输入
维护文档、维护记录、系统维护调查表;
- 测量类型
计数/计数;
- 测试方法
专家评审法;
- 实施要点:
 - 1) 确定选取的模块,并计为 B ,选取时应遵循以下原则:
 - 核心业务模块;
 - 使用频度高的模块;
 - 文档规定的模块;
 - 高优先级的模块。
 - 2) 针对所选取的模块,检查对应的设计文档,通过评审确定文档描述与选取模块的功能一致的个数,并计为 A ;
 - 3) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

7.3 数据的规范性

7.3.1 数据的规范性

数据的规范性测试要求如下:

- 前置条件
用户文档已经定义数据类型的规范;
- 测试输入
维护文档、维护记录、系统维护调查表;
- 测量类型
计数/计数;
- 测试方法
专家评审法、技术测试方法;
- 实施要点:
 - 1) 查看软件设计文档,确定软件的数据种类,并预定义数据类型规范;
注: xml schema 是一种常用的数据类型规范定义方式,为 XML 文档提供丰富的语法结构。XML Schema 对类型继承提供了广泛的支持,允许重用以前定义的结构。使用 facets,可以派生新的类型,表示其他某个类型值的更小子集,比如通过枚举、范围或模式匹配来定义子集。
 - 2) 可从以下几个方面(但不限于)考虑,但不限于统计系统与软件数据类型规范数总量,并计为 B :
 - 可出现在文档中的元素;
 - 可出现在文档中的属性;
 - 哪个元素是子元素;
 - 子元素的次序;
 - 子元素的数目;
 - 元素是否为空,或者是否可包含文本;
 - 元素和属性的数据类型;

- 元素和属性的默认值以及固定值；
- 3) 对于利用带有自动数据规范性校验功能的数据表达,例如 XML,宜采用自动校验工具进行数据类型的规范性分析。对于不具有自动校验功能的数据规范规定,宜采用专家评审法分析软件的所有数据类型;
注:在 JAVA、c# 等各种编程语言中,一般也提供了对于 XML 数据的校验手段。
- 4) 根据设计文档,统计符合规范格式的数据类型,并计为 A;
- 5) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

8 易改变性

8.1 可修改性

8.1.1 代码的可修改性

代码的可修改性测试要求如下:

- 前置条件
维护者已经完成维护,并形成了有效的维护记录;
- 测试输入
需求规格说明书、维护需求、维护计划、维护记录;
- 测量类型
计数/计数;
- 测试方法
专家评审法;
- 实施要点:
 - 1) 审阅维护计划、运行报告、测试文档、问题修改报告等相关文档;
 - 2) 根据上述文档确定需维护的功能列表,统计功能点个数,并计为 B;
 - 3) 对计划维护的修改好的功能点逐一进行测试或验证,统计测试只能通过修改代码实现维护的功能点,并计为 A;
 - 4) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

8.1.2 可配置性

可配置性测试要求如下:

- 前置条件
维护者已经完成维护,并形成了有效的维护记录;
- 测试输入
需求规格说明书、维护需求、维护计划、维护记录;
- 测量类型
计数/计数;
- 测试方法
专家评审法;
- 实施要点:
 - 1) 审阅维护计划、运行报告、测试文档、问题修改报告等相关文档;
 - 2) 根据上述文档确定需维护的功能列表,将功能点个数,计为 B;
 - 3) 对计划维护的功能点通过修改(配置)参数进行维护,完成维护的功能点计数为 A;

注 1: 常见的维护配置包括: 数据字典、权限设置、流程定义、专用的配置文件等。

注 2: 例如, 权限功能: 根据系统设置的安全规则或者安全策略, 用户可以访问而且只能访问自己被授权的资源。系统管理员只需在系统管理模块中通过配置参数即可实现不同权限管理。

注 3: 例如, 流程调整功能。在一个工作流系统中, 由于内部管理机制和部门架构的调整, 需要变更工作流的流转路径, 动态的配置功能可以通过后台的调整维护功能实现流转路径的调整, 而无需修改代码。

4) 根据 GB/T 29834.2 度量公式 $X=A/B$, 计算出测量结果。

8.2 修改实施的效率

8.2.1 变更周期的效率

变更周期的效率测试要求如下:

——前置条件

维护者已经完成维护, 并形成了有效的维护记录;

——测试输入

需求规格说明书、维护需求、维护计划、维护记录;

——测量类型

计时/计数;

——测试方法

技术测试法;

——实施要点:

1) 审阅维护计划、运行报告、测试文档、问题修改报告、产品描述等相关文档;

2) 记录第 i 次发生失效的时间, 计为 Ts_i ;

3) 记录第 i 次修复失效的时间, 计为 Tr_i ;

4) 记录每次变更的过程, 将次数计为 N ;

5) 根据 GB/T 29834.2 度量公式 $Tav = (\sum_{i=1}^n Tu_i) / N$, $Tu_i = Tr_i - Ts_i$, 计算出测量结果。

注: Tu_i 指用于修复的实际时间, 不包括由于项目挂起、中断或者其他和维护无关的因素所消耗的时间。

8.2.2 修改实施的效率

修改实施的效率测试要求如下:

——前置条件

维护者已经完成维护, 并形成了有效的维护记录;

——测试输入

需求规格说明书、维护需求、维护计划、维护记录;

——测量类型

计时/计数;

——测试方法

技术测试法;

——实施要点:

1) 审阅维护计划、运行报告、测试文档、问题修改报告、产品描述等相关文档;

2) 根据上述文档, 获取第 i 次维护实施方计划开始维护的时间, 计为 Ts_i ;

3) 第 i 次变更结束, 经验证记录完成的时间, 计为 Tc_i ;

- 4) 记录每次变更的过程,将次数计为 N ;
- 5) 根据 GB/T 29834.2 度量公式 $T_{av} = (\sum_{i=1}^n T_{u_i})/N$, $T_{u_i} = T_{c_i} - T_{s_i}$, 计算出测量结果。
注 1: T_{u_i} 指用于修复的实际时间,不包括由于项目挂起、中断或者其他和维护无关的因素所消耗的时间。
注 2: 修改实施的效率和变更周期的效率最主要的区别在于都包含了原因分析和查找时间。

8.2.3 修改的复杂度

修改的复杂度测试要求如下:

- 前置条件
维护者已经完成维护,并形成了有效的维护记录;
- 测试输入
需求规格说明书、维护需求、维护计划、维护记录;
- 测量类型
计时/计数;
- 测试方法
评审法;
- 实施要点:
 - 1) 审阅维护计划、运行报告、测试文档、问题修改报告、产品描述等相关文档;
 - 2) 记录变更次数,计为 N ;
 - 3) 记录第 i 次变更所花费的时间,计为 A_i ;
 - 4) 计算第 i 次变更后该系统与软件的规模度,计为 B_i ;
- 5) 根据 GB/T 29834.2 度量公式 $T = (\sum_{i=1}^N A_i/B_i)/N$, 计算出测量结果。

8.3 修改的可控制性

8.3.1 修改的可还原性

修改的可还原性测试要求如下:

- 前置条件
维护者已经完成维护,并形成了有效的维护记录;
- 测试输入
需求规格说明书、维护需求、维护计划、维护记录;
- 测量类型
计数/计数;
- 测试方法
技术测试法;
- 实施要点:
 - 1) 审阅用户手册、产品描述、测试文档等相关文档;
 - 2) 根据上述文档获取修改后可还原至修改前状态的功能点数,并计为 B ;
 - 3) 进行还原操作并进行测试和验证,还原操作可包括:
 - 软件和系统对操作的实时 Redo/Undo 功能;
 - 对于修改,系统提供了备份功能,在修改失败或者用户需要时,还原系统维护前的状态;

- 软件提供了状态检查点功能,在需要时自动设置设置系统当前的运行状态,用户维护开始以后,使得用户有条件还原至维护前的制定状态。
- 4) 统计成功还原的功能点数,并计为 A ;
- 5) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

8.3.2 软件变更控制的能力

软件变更控制的能力测试要求如下:

——前置条件

维护者已经完成维护,并形成了有效的变更管理和维护记录;

——测试输入

变更记录、需求规格说明书、维护需求、维护计划、维护记录;

——测量类型

计数/计数;

——测试方法

技术测试法,采用变更管理的方式统计软件的变更控制能力;

——测试方法:

- 1) 审阅运行报告、维护计划、测试文档等相关文档;
- 2) 根据相关文档获取版本变更次数,计为 B ;
- 3) 通过评审变更记录,查看所有的版本是否有明确唯一的标识,记录查看结果,将有明确唯一标识的个数计为 A 。

注 1: 版本号一般由二至四个部分组成:主版本号、次版本号、内部版本号和修订号。主版本号和次版本号两个部分为必选。内部版本号和修订号两个部分为可选;但是,只有在定义内部版本号部分时,修订号部分才为可选。格式为:主版本号.次版本号[.内部版本号[.修订号]]。

注 2: 可借助版本管理工具查看软件每个版本。

- 4) 根据度 GB/T 29834.2 度量公式 $X=A/B$,计算出测量结果。

9 稳定性

9.1 变更成功的比率

变更成功的比率测试要求如下:

——前置条件

维护者已经完成维护,并形成了有效的变更管理和维护记录;

——测试输入

维护计划、维护记录、可运行的系统;

——测量类型

计数/计数;

——测试方法

评审法;

——实施要点:

- 1) 审阅用户手册、产品描述、测试文档等相关文档;
- 2) 记录在维护前的某段时间系统与软件失效的次数,计为 B ;
- 3) 记录在维护后的某段时间系统与软件失效的次数,计为 A ;
- 4) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算测量结果。

9.2 修改影响的局部化

修改影响的局部化测试要求如下：

——前置条件

维护者已经完成维护,并形成了有效的变更管理和维护记录；

——测试输入

维护计划、维护记录、可运行的系统；

——测量类型

计数/计数；

——测试方法

技术测试法；

——实施要点：

- 1) 审阅用户手册、产品描述、测试文档等相关文档；
- 2) 记录解决的失效数,计为 N ；
- 3) 记录在特定的时期内通过变更解决失效后再次出现的失效数,计为 A ；
注：时间的长短宜根据软件的领域特性决定。
- 4) 根据 GB/T 29834.2 度量公式 $X=A/N$,计算测量结果。

10 可验证性

10.1 可自动验证性

可自动验证性测试要求如下：

——前置条件

维护者已经完成维护,并形成了有效的变更管理和维护记录；

——测试输入

维护计划、维护记录、可运行的系统；

——测量类型

计数/计数；

——测试方法

运行自动验证模块；

——实施要点：

- 1) 审阅用户手册、产品描述、测试文档等相关文档；
- 2) 根据系统与软件的维护相关文档(如软件需求设计文档),确定系统与软件实际需要进行维护的功能个数,计为 B ；
- 3) 维护操作结束之后,适用时设置自动验证功能,启动并运行该软件；
- 4) 依据相关文档,观察系统与软件是否已经进行了自动验证。自动验证的证据可包括：
 - 软件界面是否有维护记录(例如,被更改的功能点清单以及经过自动验证,该功能点是否被更改成功)；
 - 或者考察日志是否记录了自动验证的过程及详细内容；
 - 或者统计界面上已列举的已验证功能点清单的个数；
- 5) 已自动验证的功能点个数,计为 A ；
- 6) 根据 GB/T 29834.2 度量公式 $X=A/B$,计算测量结果。

示例：

若软件能具备自动验证的功能，则度量值为高。

某软件计划对 10 个(B)功能点进行修改，维护后，重新运行该软件，软件启动时弹出信息框记录验证过程，列举被修改的功能清单以及维护结果(成功/失败)，统计信息框中已列举的验证成功的功能点个数 10 个(A)，则自动验证值为： $X=10/10=1$ 。

10.2 测试的重启性

测试的重启性测试要求如下：

——前置条件

维护者已经完成维护；

——测试输入

维护计划、维护记录、可运行的系统；

——测量类型

计数/计数；

——测试方法

技术测试法；

——实施要点：

- 1) 审阅用户手册、产品描述、测试文档、维护文档等相关文档；
- 2) 记录在测试中暂停测试的总次数，计为 B ；
- 3) 记录在测试中能暂停并重新开始执行测试的事例数，计为 A ；
- 4) 根据 GB/T 29834.2 度量公式 $X=A/B$ ，计算测量结果。

10.3 维护完整性

维护完整性测试要求如下：

——前置条件

维护者已经完成维护；

——测试输入

维护计划、维护记录、可运行的系统；

——测量类型

计数/计数；

——测试方法

专家评审法；

——实施要点：

- 1) 审阅用户手册、产品描述、测试文档、维护文档等相关文档；
- 2) 根据上述文档，确定系统与软件实际需要进行维护的功能点个数，计为 B ；
- 3) 在实际的维护操作中，邀请专家对评审由于现有技术条件、成本限制、时间考虑等因素未进行修改的功能；
- 4) 统计已维护成功的功能点个数进行统计，计为 A ；
- 5) 根据 GB/T 29834.2 度量公式 $X=A/B$ ，计算测量结果。

附录 A
(资料性附录)
代码分析示例

本附录给出代码可维护性分析示例。

a) 系统的概要

图 A.1 给出了某一个系统的代码规范性样例。包含了代码行数、注释比例、测试代码覆盖率、规范比例等四个部分。

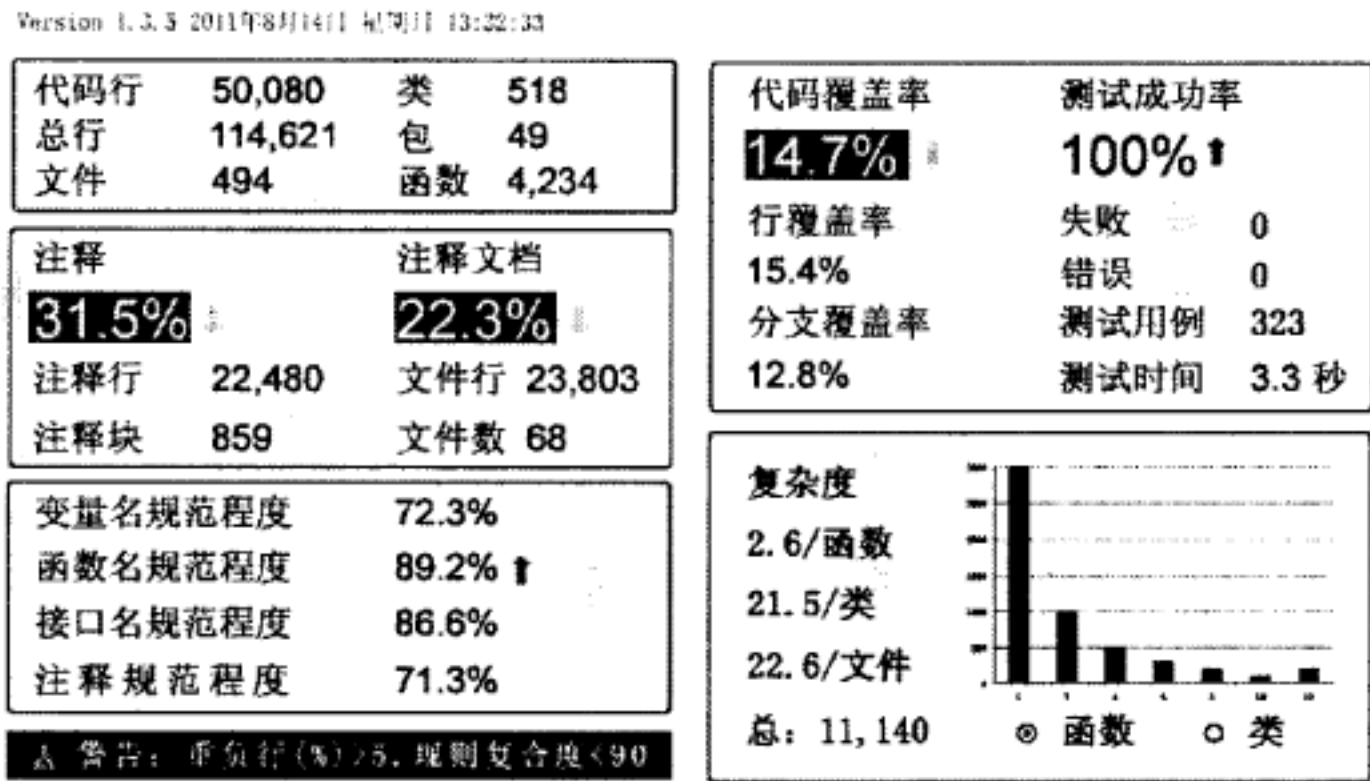


图 A.1 某系统的代码规范性概要

b) 代码的规范性检查

以某文件为例,给出了相关的维护性指标,如表 A.1 所示。

表 A.1 规范性检查结果示例

代码规范指标	值	规范指标	值
行数	100	注释比例 (%)	24.6%
代码行数	46	注释行数	15
方法数	2	类数	1
语句数	20	集成的子类数	0
复杂性	5	继承树深度	3
平均每个方法的复杂性	2.5	响应该类的类数	24
共用 API	1		—

c) 违反预定义的规则示例

表 A.2~表 A.4 分别为几个规则违反的例子,包括常量名规则违反、不必要的括号、具体申明抛出的申明等。

表 A.2 常量名规则违反

规则名称	常量名
含义	检查遵守特定格式的常量名
所在位置(行号)	45
代码	private static final Log log = LogFactory.getLog(CreateAction.class);
在代码中表现	常量名'log'必需匹配以下规则: '[A-Z][A-Z0-9]*(_[A-Z0-9]+)*\$'

表 A.3 具体声明抛出的异常

规则名称	具体声明抛出的异常
含义	由于无法肯定何种类型的异常将会被抛出,因此定义和理解模糊接口较为困难。建议使用一个继承自 RuntimeException 的派生类或者其他方法来检查异常
所在位置(行号)	52
代码	protected synchronized Action getAction(ActionContext context, String type, ActionConfig actionConfig) throws Exception {
在代码中表现	方法或函数不应明确的抛出异常:java.lang.Exception

表 A.4 不必要的括号

规则名称	不必要的括号
含义	有时表达式包含在不必要的括号中,使它们看起来像一个函数调用
所在位置(行号)	80
代码	return (action);
在代码中表现	语句可能有一些不必要的括号

参 考 文 献

[1] GB/T 5271.1- 2000 信息技术 词汇 第 1 部分:基本术语(eqv ISO/IEC 2382-1:1993)

[2] GB/T 5271.20—1994 信息技术 词汇 20 部分:系统开发(eqv ISO/IEC 2382-20:1990)

[3] GB/T 8566—2007 信息技术 软件生存周期过程

[4] GB/T 14079 1993 软件维护指南

[5] GB/T 16260.1—2006 软件工程 产品质量 第 1 部分:质量模型(ISO/IEC 9126-1:2001, IDT)

[6] GB/T 16260.2 2006 软件工程 产品质量 第 2 部分:外部度量(ISO/IEC TR 9126-2: 2003,IDT)

[7] GB/T 16260.3—2006 软件工程 产品质量 第 3 部分:内部度量(ISO/IEC TR 9126-3: 2003,IDT)

[8] GB/T 16260.4—2006 软件工程 产品质量 第 4 部分:使用质量度量(ISO/IEC TR 9126-4:2004,IDT)

[9] GB/T 18905.2—2002 软件工程 产品评价 第 2 部分:策划和管理(ISO/IEC 14598-2: 2000,IDT)

[10] GB/T 18905.3—2002 软件工程 产品评价 第 3 部分:开发者用的过程(ISO/IEC 14598-3:2000,IDT)

[11] GB/T 18905.4—2002 软件工程 产品评价 第 4 部分:需方用的过程(ISO/IEC 14598-4:1999,IDT)

[12] GB/T 18905.5—2002 软件工程 产品评价 第 5 部分:评价者用的过程(ISO/IEC 14598-5:1998,IDT)

[13] GB/T 18905.6—2002 软件工程 产品评价 第 6 部分:评价模块的文档编制(ISO/IEC 14598-6:2001,IDT)

[14] GB/T 19000—2008 质量管理体系 基础和术语(ISO 9000:2005,IDT)

[15] GB/T 19001—2008 质量管理体系 要求(ISO 9001:2008,IDT)

[16] GB/T 20157—2006 信息技术 软件维护(ISO/IEC 14764:1999,IDT)

[17] ISO/IEC 25010:2011 Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—System and software quality models

