

中华人民共和国金融行业标准

JR/T 0130—2016

银行业软件异常分类

Classification for bank's software anomalies

2016 - 07 - 01 发布

2016 - 07 - 01 实施

中国人民银行

发布

目 次

前言	II
引言	III
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 模型	2
4.1 缺陷生命周期	2
4.2 异常实体关系模型	2
4.3 异常相关实体分类过程与本标准的关系	4
5 分类	5
5.1 分类过程	5
5.2 缺陷分类属性	5
5.3 失效分类属性	6
附录 A (资料性附录) 属性示例值	7
附录 B (资料性附录) 分类范例	11
B.1 意图	11
B.2 问题场景描述	11
B.3 分类示例	11
参考文献	14

前 言

本标准按照国家标准GB/T 1.1—2009给出的规则起草。

本标准由中国农业银行股份有限公司提出。

本标准由全国金融标准化技术委员会（SAC/TC180）归口。

本标准起草单位：中国农业银行股份有限公司，中国人民银行金融信息中心，中国金融电子化公司，中国工商银行股份有限公司。

本标准主要起草人：刘国建、于进、叶又升、潘贵平、李宽、吴俊峰、伞亮、侯太利、王子田、李越、何涛、吴金超、夏法鹏、吴晓光、张雯华、时向一、仲海港、魏猛。

引 言

“异常”一词可指任何不正常、不规则、不一致或者对预期的偏离。异常可以指状态或事件，可以指表现或行为，也可以指形式或功能。本标准对软件异常的分类提供了一种一致的方法，以分类在项目、产品或系统生命周期的任何阶段产生并在后继的任何阶段发现的软件异常。异常分类数据可以用于多种用途，包括缺陷原因分析、软件过程改进（如减少缺陷嵌入的可能性或者提高缺陷早期检测的可能性）等。

收集本标准中所描述的数据能为提高IT管理水平和增进软件质量提供宝贵信息。一是在软件生命周期中越早发现问题，修改成本越低，通常也越容易。这就推动着使用各种工具、技术和方法来更早地发现问题。标准的异常数据在评估这些工具、技术和方法的效用时可作为重要参考。二是通过对异常数据的分析，也能发现项目生命周期中产生问题最多的阶段。明辨软件中的提高和问题，对软件异常的修正优先级判定和投入资源的分配提供了有价值的参考信息。三是异常数据能辅助对可靠性和生产率等质量指标的评估。

形成对软件异常的标准分类有着重要的工程意义。

首先，它能使本标准的应用者深刻理解产品开发过程中所产生的异常类型，这些异常信息是项目执行以及过程改进的重要数据来源。诸如正交缺陷分类和因果分析法之类的分析型技术，能够以异常分类来识别缺陷的根本原因以避免缺陷重现。过程改进框架（如能力成熟度模型CMMI®）则增大了详细理解过程性能和产品质量的需要。异常分类让对各种开发过程中产生的异常成为评判过程优劣的一个指标。

其次，采用标准的方法对软件异常进行分类，能让开发者和组织之间更好地传达、交流关于异常的信息。遗憾的是，人们常常赋予同一个词不同的意思或者使用不同的词来表达同一个意思。同样，如果软件系统之间要有效沟通关于异常的信息，就必须使用共同的逻辑数据模型（如果不要求共同物理数据模型的话）。如果同样的数据元素在一个系统和其他系统中命名不同，通过一些映射和转译方法仍然能实现数据交换，但是每个系统都必须至少识别并实现同样的概念对象/实体、关系和属性。

本标准的目的是定义一套通用的词汇，以使得银行业相关人员、组织能够就软件异常进行有效的沟通；同时建立一套软件缺陷和失效数据的通用属性集，以便采用工业技术进行分析。

银行业软件异常分类

1 范围

本标准对银行业软件失效和缺陷的分类规定了一组核心属性。对特定的应用程序或业务需求来说，还会存在有特定价值的其他失效或缺陷的属性。本标准未涉及生命周期模型的选择，但选择任何生命周期模型均不影响分类属性。本标准应用者可基于所选择的生命周期对分类属性值进行裁剪。问题分类、错误分类、变更请求、检测和移除缺陷、研究和解决失效的方法和过程，以及确定缺陷是否应该移除的过程均不在本标准的范围中。

本标准适用于对任何银行业软件（包括操作系统、数据库管理系统、应用程序、测试软件、固件和嵌入式软件）的失效和缺陷进行分类。

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，仅所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—2006 信息技术 软件工程术语

ISO/IEC/IEEE 24765:2010 系统和软件工程 词汇 (Systems and software engineering — Vocabulary)

3 术语和定义

GB/T 11457—2006和ISO/IEC/IEEE 24765:2010中确立的以及下列术语和定义适用于本文件。

3.1

缺陷 defect

在一个项目组件中的瑕疵或缺点，导致该组件不能满足其需求或规格，且需要修复或更换。

[ISO/IEC/IEEE 24765:2010, 3.764]

注：在ISO/IEC/IEEE 24765:2010中，除上述定义外，还有两个定义，分别为：“一个若不更正则可能引起一个应用失败，或者产生不正确结果的问题。”引自ISO/IEC 20926:2003 《Software engineering — IFPUG 4.1 Unadjusted functional size measurement method—Counting practices manual》；“一个涉及到故障（原因）或失效（结果）的通用术语。”，引自IEEE Std 982.1—2005 《Dictionary of Measures of the Software Aspects of Dependability》。

示例1：在项目生命周期的早期阶段发现的遗漏和瑕疵。

示例2：在足以进行测试或操作的软件中存在的故障。

3.2

错误 error

产生不正确结果的人的行为，不正确结果如软件包含一个故障。

[ISO/IEC/IEEE 24765:2010, 3.1027]

注：在ISO/IEC/IEEE 24765:2010中，除上述定义外，还有3个定义，分别为：“不正确的步骤、过程或数据定义。”

“不正确的结果。” “在计算的、观察的或测量的值或情况与准确的、规定的或理论上正确的值或情况之间的差异。”

示例：在软件规格中对用户需求的遗漏和误解；在设计规格中对需求的遗漏或不正确地翻译。

3.3

失效 failure

系统或系统组件未在规定范围内执行所需功能的事件。

[ISO/IEC/IEEE 24765:2010, 3.1115]

注：故障发生时可能会引起失效。

3.4

故障 fault

软件中错误的一种表现。

[ISO/IEC/IEEE 24765:2010, 3.1122]

注1：ISO/IEC/IEEE 24765:2010 中，除上述定义外，还有 2 个定义，分别为：“在一个计算机程序中，一个不正确的步骤、过程或数据定义”。“在一个硬件设备或部件中的缺陷”。

注2：如果遇到一个故障，则可能引起一个失效。

3.5

问题 problem

一人或多人由于系统使用中的不符合要求而经历的困境或不确定性。

注：在ISO/IEC/IEEE 24765:2010中，还有2个定义，分别为：“一个或多个事件的未知的根本原因”。“需克服的不利状况”，这一定义也描述于IEEE Std 1044—2009中。

4 模型

4.1 缺陷生命周期

图1所示为缺陷的生命周期。由于缺陷在检测出之前无法进行分类，故本标准针对已检测出的缺陷分类，并对表明了存在缺陷的失效分类。



图1 用 UML 状态图描述的缺陷生命周期

4.2 异常实体关系模型

发现问题是对失效进行识别的前提，这些问题是用户识别出的软件在不恰当方式下执行的状况。同样的，对失效和故障所采取的措施可能会被记录为一个变更请求。

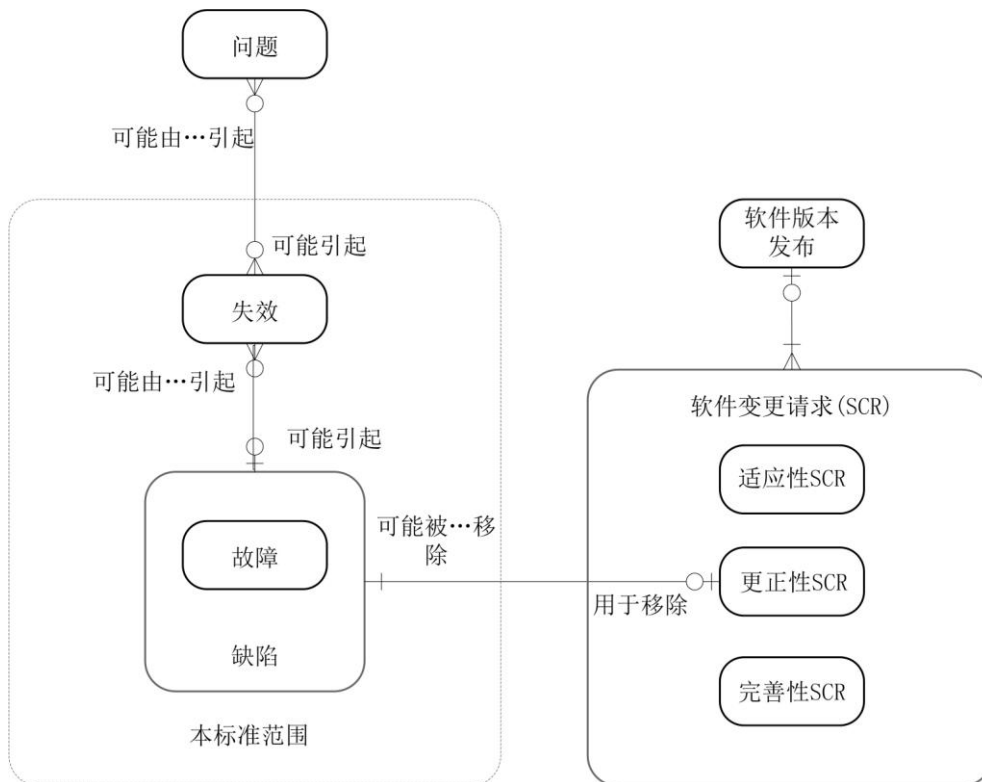
尽管软件变更请求（SCR）和软件版本在本标准内未提出，但它们也出现在图2、图3以及表1中，以帮助阐明范围。这里展示的是相对简单的情况，一个缺陷有可能和一个修正SCR有关联，且各个SCR有可能关联在一起，但至多只能存在于一个软件发布版本中。

表1采用文字方式描述了对异常相关实体间关系的说明。

表1 对异常相关实体间关系的说明

类/实体对	关系
问题—失效	一个问题可能由一个或多个失效引起。 一个失效可能会引起一个或多个问题。
失效—故障	一个失效可能由一个故障引起（失效亦表明了故障的存在）。 一个故障可能引起一个或多个失效。
故障—缺陷	故障是父类缺陷的子类。 任一个故障都是缺陷，但并不是任一个缺陷都是故障。 若缺陷是在软件执行过程中遇到的（由此导致失效），则该缺陷是故障。 若缺陷是通过审查或者静态分析发现的，且在执行软件之前被移除，则该缺陷不是故障。
缺陷—变更请求	缺陷可以通过执行更正性变更请求移除。 更正变更请求的目的就是移除缺陷。 (变更请求也可以通过执行自适应或完善式维护的方式启动)

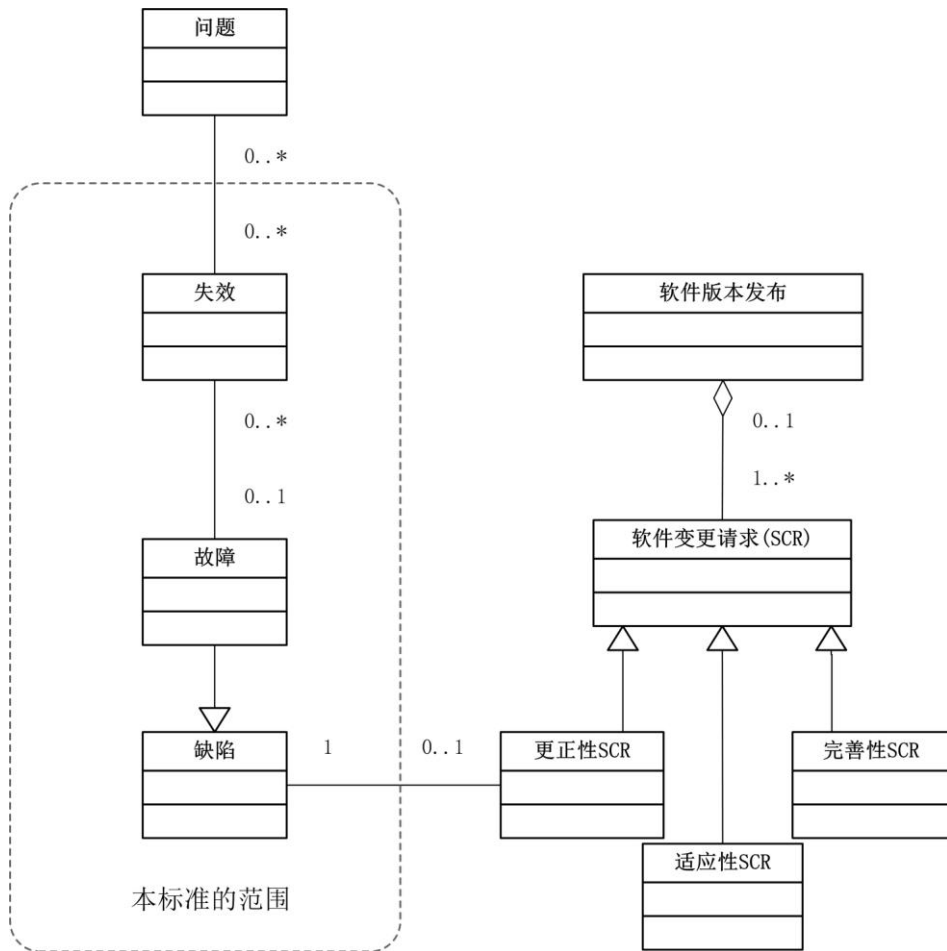
图2和图3分别用实体-关系图和UML的类图描述了这些实体间的关系，这两种描述方式表述的内容是一致的。



注1：圆角矩形代表实体(关注物)，连接圆角矩形的线代表实体之间的关系。线末端的符号表明实体的数目。线末端附近的圆圈表明实体数目允许为零，即参与是可选的；当没有圆圈时表明至少需要一个实体，即参与是必需的。三条腿的“鸡爪”符号表明允许多个实体参与；当没有鸡爪符号时表明至多有一个实体可以参与。一个圆角矩形出现在另一个圆角矩形中表明父子类关系，此处被包含的实体可归为包含实体（父类型）的子类型。

注2：此图的目的是不是制定强制的符号标识方法，也不作为数据库的结构描述。

图2 异常相关实体的实体-关系模型



注1：矩形代表对象类(关注物)，连接矩形的线代表对象类之间的关系。每个矩形内的三个部分(从上到下)包含相应类的名称、属性和方法。该图的主要焦点是类之间的关系，因此图中仅包含类名。“方法”在本标准的范围以外，属性的列表及定义在下面的章节中。线旁边的数字表明关系的多重性，“1”的意思是只有一个，“0..1”的意思是零个或一个，“1..*”的意思是一个或多个，“0..*”的意思是零个、一个或更多。末端为三角形的线表明父类与子类之间的一般化关系（父类-子类）。末端为菱形的线表明一个版本中可能会包含的多个更改请求。

注2：此图的目的是不是制定强制的符号标识方法，也不作为数据库的结构描述。

图3 异常相关实体的UML类图

4.3 异常相关实体分类过程与本标准的关系

表2给出了关于异常相关实体的分类过程哪些在本标准范围内，哪些在本标准范围外。

表2 异常相关实体的分类过程与本标准的关系

本标准范围内	本标准范围外
缺陷分类	修正措施分类
故障分类	错误分类
失效分类	问题分类
定义一组广泛适用于分类的核心属性	定义所有有可能有用的分类属性
定义模板属性值以促进理解	裁剪模板属性以满足特定组织的需求

表 2（续）

本标准范围内	本标准范围外
	在项目或产品生命周期中创建正式的分类过程
	定义一个过程，规定哪个值应该分配给哪个属性
	定义一个过程，规定何时何地何人如何记录属性值
	是否移除缺陷的处理过程

5 分类

5.1 分类过程

组织应定义其分类过程如下：

- a) 明确通过分类缺陷和失效要实现的目标；
- b) 明确用于判断出现怎样的系统/软件行为就构成失效的准则(如描述于规格、合同或计划中的)；
- c) 明确如何解决分类决策的分歧或冲突；
- d) 明确在项目或产品的生命周期内，分类何时开始与结束；
- e) 明确项目、产品或者组织特有的值，这些值适宜被指派为分类属性（属性值示例参见表 A.1 和表 A.2）；
- f) 明确由谁针对每个发现的缺陷和失效向表 3 和表 4 中所列的分类属性赋值；
- g) 明确在哪里和怎样维护分类数据。

注：分类过程可根据企业的管理惯例，采用企业标准、规章制度或非企业标准非规章制度的技术规范方式描述。

5.2 缺陷分类属性

组织应记录表3中所列所有缺陷属性的值。属性集并不试图一个不漏。表A.1为所选属性值的示例，这些值仅是资料性的。在组织处理缺陷时，属性的值很可能同时被一并记录下来，缺陷的状态可以是嵌入、检测或者移除（如图1）；然而，相比跟踪缺陷的状态，跟踪相关缺陷报告的状态更为普遍，因此这些值描述的状态通常是在缺陷解决过程中相关的组织工作流程的状态。由于存在多种组织特定的缺陷工作流程，故未指定强制的状态值。

表3 缺陷分类属性

属性	定义
缺陷 ID	缺陷的唯一标识符。
描述	有关缺失了什么、错误或多余东西的描述。
状态	缺陷报告生命周期中的当前状态。
资产	包含缺陷的软件资产（产品、组件、模块等）。
工件	包含缺陷的特定软件工作产品。
版本检测	检测出缺陷时的软件版本标识。
版本修正	缺陷被修正时的软件版本标识。
优先级	缺陷的处理顺序，该顺序由负责缺陷评估、解决和关闭的组织通过与其他已经报告的缺陷相比而给出。
严重性	缺陷可能（或曾经）引起的最严重的失效影响，该影响由负责软件工程的组织（从其视角）确定。
概率	该缺陷再次引起失效的可能性。

表3 (续)

属性	定义
影响	由缺陷引起的失效而影响到的需求类别。
类型	一种分类方式，按照发现缺陷的代码或发现缺陷的工作产品进行划分。
模式	一种分类方式，按照缺陷由不正确的实现或表示、需求外的多余添加或遗漏而进行划分。
嵌入活动	在缺陷被引入或嵌入期间（例：在包含缺陷的工件产出过程中）的活动。
检测活动	在缺陷被检测到期间的活动（即检查或测试）。
检测日期	缺陷被检测到的日期。
修正日期	缺陷被修正时的日期。
关联失效	由缺陷引起失效的标识符。
关联变更	为了修正缺陷所提出的修正变更请求对应的标识符。
处置	在关闭时缺陷报告的最终处置。

5.3 失效分类属性

组织应记录表4中所列全部失效属性的值。所选属性值的示例见表A.2。

表4 失效分类属性

属性	定义
失效 ID	失效的唯一标识符。
状态	在失效报告生命周期中的当前状态。
标题	为进行总结报告而对失效的简明描述。
描述	对异常行为及其出现条件的全面描述，包括事件顺序和[或]失效前的用户行为。
环境	观察到失效的操作环境的标识。
配置	包括相关产品和版本标识符的配置细节。
严重性	由负责软件工程的组织（从其视角）来决定。示例参见表 B.1。
分析	通过失效调查结论的因果分析的最终结果。
处置	失效报告的最终处置。示例参见表 B.1。
观察人	观察到失效的人员（可以获得额外的细节的人员）。
开启人	开启（提交）失效报告的人员。
负责人	负责调查失效原因的人员或组织。
关闭人	关闭失效报告的人员。
观察日期	观察到失效的时间/日期。
开启日期	失效报告开启（提交）的时间/日期。
关闭日期	失效报告关闭和分派最终处置的时间/日期。
关联测试	当失效发生时进行的特定测试（如存在）的标识。
关联事件	若失效报告是从服务台或帮助台的请求/沟通中提炼，相关事件的标识。
关联缺陷	声称引起失效的缺陷的标识。

附 录 A
(资料性附录)
属性示例值

本附录给出了缺陷属性值和失效属性值的一些例子，这些例子可能是常见的，本标准的应用者可以以这些例子作为初始值，归纳适应本组织的典型属性值；也可针对不同性质的软件，给出几套不同的典型取值。

表A.1 缺陷属性值示例

属性	值	定义
状态	打开	对一个检测到的缺陷的反应，后继的行动是准备好的。
状态	关闭	无进一步行动计划（无论是否已将缺陷移除）。
优先级	高	优先级为该等级的缺陷将优先分析解决。
优先级	中	优先级为该等级的缺陷将在队列中等候分析解决，且一直排在在高优先级之后。
优先级	低	具有该等级的缺陷在将在等待队列未分析解决。
严重性	阻塞	直到改正或适宜的应急方案启用，检测被阻止或暂停。
严重性	关键性	基础操作不可避免地中断，安全受到损害与威胁。
严重性	较大	基础操作受到影响，但可继续。
严重性	较小	非基础性操作中断。
严重性	不重要	未对操作产生实质性影响。
概率	高	发生的概率大于 70%。
概率	中	发生的概率介于 40%到 70%之间。
概率	低	发生的概率小于 40%。
影响	功能性	实际或潜在原因，由于该原因未能正确执行必须功能（或实现非必需的功能），包括任何影响数据完整性的缺陷。
影响	可用性	实际或潜在原因，由于该原因未能满足可用性（易用性）需求。
影响	安全性	实际或潜在原因，由于该原因未能满足安全性需求，诸如鉴别、授权、隐私/机密、可计量性（如审计跟踪或事件日志）等。
影响	性能	实际或潜在原因，由于该原因未能满足性能需求（如容量、计算精度、响应时间、吞吐量或可用性）。
影响	服务能力	实际或潜在原因，由于该原因未能满足可靠性、可维护性、可支持性要求（如复杂的设计，无文档代码，有歧义或不完整的错误日志等）。
影响	其他	将不会导致上述结果的影响。

表 A.1 (续)

属性	值	定义
类型	数值	<p>在数据定义、初始化、映射、访问或使用中的缺陷，这些缺陷是在模型、规格或实施中发现的。</p> <p>例如：</p> <ul style="list-style-type: none"> ①未分配初始值或未设置标志的变量。 ②不正确的数据类型或列的大小。 ③使用不正确的变量名。 ④未定义有效范围。 ⑤数据模型中不正确的关系基数，如 1..n 标记为 0..n。 ⑥选择列表中值缺失或有不正确的值。
类型	接口	<p>在接口规格或实现中的缺陷（如用户和机器之间，内部两个软件模块之间，软件模块和数据库之间，内部和外部软件组件之间，软件和硬件之间等）。</p> <p>例如：</p> <ul style="list-style-type: none"> ①不正确的模块接口设计或实现。 ②不正确的报表布局（设计或实现）。 ③不正确或不充分的参数传递。 ④用户界面中隐藏的或陌生的标签或消息。 ⑤发送或显示不完整或错误的消息。 ⑥数据输入屏上缺少必须的区域。
类型	逻辑	<p>决策逻辑、分支、排序或算法中的缺陷，这些缺陷在自然语言规格或实现语言中被发现。</p> <p>例如：</p> <ul style="list-style-type: none"> ①缺少 else 语句。 ②不正确的操作顺序。 ③表达式中不正确的运算符或操作数。 ④缺少对返回响应进行状态检测的逻辑（如返回代码、文件结尾，空值等）。 ⑤输入值不在有效范围内。 ⑥序列图中缺少系统响应。 ⑦需求分析或设计规格中业务规则的歧义定义。
类型	描述	软件描述、使用、安装或操作中的缺陷。
类型	语法	不符合语言定义的规则。
类型	标准	不符合定义的标准。
类型	其他	未定义类型的缺陷。
模式	错误	不正确、不一致或不明确的东西。
模式	丢失	应该存在的东西缺失。
模式	额外	不需要的东西存在。

表 A.1 (续)

属性	值	定义
嵌入活动	需求	在需求定义活动中嵌入的缺陷（如需求创意、分析或规格中嵌入的缺陷）。 例如： ①需求规格中遗漏满足客户目标所需的功能。 ②不完整的用例规格。 ③缺失或不正确的性能需求。 ④缺失或不正确的安全性需求。 ⑤需求规格中未正确规定的功能。 ⑥需求规格中满足客户目标并不需要的功能。
嵌入活动	设计	设计活动期间嵌入的缺陷。 例如： ①设计无法支持所提出的需求。 ②从逻辑数据模型中未正确推导出物理数据模型。 ③不正确的应用程序接口设计。
嵌入活动	编码	在“编码”或类似活动中嵌入的缺陷。 例如： ①不正确的变量类型。 ②不正确数据的初始化。 ③模块接口未按设计编码。
嵌入活动	配置	在产品构建或封装过程中嵌入的缺陷。 例如： ①构建过程中包含的错误源文件。 ②分发/部署程序包中包含的错误. EXE 文件。 ③在. INI 文件中有错误的本地化参数。
嵌入活动	文档	在安装或操作说明文档中嵌入的缺陷。 例如： ①用户手册中列出的不正确的菜单选项。 ②在线帮助中不正确的任务或导航指示。 ③产品规格中缺少必要的安装条件。 ④产品版本注释中错误的版本标识符。
检测活动	需求	在需求整合、检查、评审过程中检测到的缺陷。
检测活动	设计	在设计整合、检查、评审过程中检测到的缺陷。
检测活动	编码	在源代码整合、检查、评审过程中检测到的缺陷。
检测活动	供应商测试	在任何由供应商进行的测试中检测到的缺陷。
检测活动	客户测试	在任何由客户进行的测试中检测到的缺陷。
检测活动	生产	在生产运行和使用中检测到的缺陷。
检测活动	审计	审计中检测到的缺陷（发行前或发行后）。
检测活动	其他	在任何其他活动中检测到的缺陷，如用户/操作员的培训或产品演示过程中检测到的缺陷。
处置	修正	修正/移除缺陷。
处置	未找到	未找到缺陷。失效无法复现，或报告的行为实际上是预期的行为。

表 A.1 (续)

属性	值	定义
处置	参考	在另一组织的资产中到的缺陷，且缺陷由该组织修正。
处置	重复	重复报告的缺陷。

表A.2 失效属性值示例

属性	值	定义
状态	打开	未来的行动是可预期的。
状态	关闭	没有进一步行动的计划。
严重性	关键	不可避免地中断基本操作和[或]危及安全。
严重性	较大	影响基本操作，但能继续。
严重性	较小	中断非基本操作操作。
严重性	无关紧要	对操作无显著影响。
处置	未知原因	未找到失效原因；失效征兆消失了。
处置	重复	同一失效事件在另一份报告中已经存在。
处置	解决	发现失效原因并解决。

附 录 B

(资料性附录)

分类范例

B.1 意图

本附录给出了5个银行业应用软件问题场景的范例，意图通过这些范例中对缺陷和失效指标的确定，为本标准的应用提供参考。

B.2 问题场景描述

范例问题场景如下所列，其相关的分类数据在表B.1和表B.2中给出。

场景 1：甲银行新开发了一个低柜管理系统，小赵是该银行的客户经理。在系统上线后，小赵发现在登陆屏上找不到密码输入框，因而无法登陆低柜管理系统，小赵在查询了培训笔记后，也没有发现对这个问题的专门记载，所以焦急地向上级主管部门反映这一问题。

分析：小赵遇到的不能登录低柜管理系统的问题，是由于因密码输入框未在登陆屏上显示而导致的故障，进一步地是由于在 Login.asp 工件编码过程中嵌入的缺陷造成的。

场景 2：甲银行低柜管理系统的技术支持团队正在与同一银行另一个网点的小钱沟通，小钱也遇到了类似的问题而无法登陆该低柜管理系统。

分析：小钱遇到的问题与小赵的相似，足以说明两个不同的故障（事件）可以由一个缺陷（条件）引起。

场景 3：乙银行开发的智慧网点系统组织用户合格性测试，测试人员小孙发现字体的颜色不符合需求文档中 4.2.1.3 节的要求。

分析：该范例说明了故障（屏幕上不正确颜色的出现）和引起该故障的缺陷（在代码中将不正确的数据值赋给一个常量）的差异。

场景 4：丙银行组织了新一代财务管理系统的软件需求同行评审，老李发现在需求中的值是以分为单位而不是以元为单位。

分析：该范例说明在任何故障发生前，应先对直接检测到的失效进行分类。

场景 5：丁银行的一个网点所处的地段平常并不停电，但依旧配备了 UPS。在一个因电路故障导致的突然停电中，丁银行该网点的电池耗尽了电量，并导致了最后一笔业务的状态不能确定，经过电话联系数据运行中心进行查询，才确定了最后一笔业务的状态，并向客户再三解释、致歉，才没有导致客户的投诉。丁银行该网点的负责人老周十分恼火，在上级行来对 IT 支持情况进行调研时，向调研人员吴主任反应了这个情况。吴主任在回来后，对这一问题进行了彻查，发现在提出的需求中指定要在电池电量能够支持 10 分钟和 2 分钟时，分别报警，以便疏导客户和执行终端设备下电操作，但在网点没有专门的安全监控系统，故未包括能够满足这一需求的电池电量不足警告。

分析：该范例中，一直到在生产环境出现失效前，缺陷都没有被检测出来。

B.3 分类示例

表B.1给出了各场景的失效属性值。

表B.1 失效分类示例

失效属性	场景 1	场景 2	场景 3	场景 4	场景 5
失效 ID	F080001	F080002	FID001	不适用	DID005
标题	失效：缺失密码区域	失效：缺失密码区域-重复	显示 - 错误的字体颜色	不适用	缺少电池电量不足警报
描述	登陆屏缺失密码域	登陆屏缺失密码域	字体颜色不符合系统规格：黑色而不是蓝色	不适用	银行网点缺失安全监控系统，导致没有电池电量不足警报
环境	总行 WebServer-2	总行 WebServer-2	Windows XP	不适用	银行网点环境管理系统
配置	低柜业务系统 1.0	低柜业务系统 1.0	显示 - 版本 -4.0	不适用	银行网点电源监控模块
严重性	关键	关键	较小	不适用	关键
分析	代码错误		字体颜色不符合规范中 4.2.1.3 节的要求	不适用	网点环境监控设计未包括电池电量低
处置		重复		不适用	
观察人	小赵	小钱	小孙	不适用	老周
开启人	郑工程师	王工程师	冯分析师	不适用	吴主任
负责人			陈工程师	不适用	
关闭人					楮工程师
观察日期	2012.5.5	2012.5.6	2013.6.5	不适用	2006.5.8
开启日期	2012.5.5	2012.5.6	2013.6.7	不适用	2006.5.10
关闭日期				不适用	2006.10.15
关联测试	不适用	不适用	Disp_font_ver_1	不适用	Securemon 09
关联事件	S080002	S080003	HD001	不适用	SEC054
关联缺陷	D080234	D080234	F080001	不适用	SD089

表 B.2 给出了各场景的缺陷属性值。

表B.2 缺陷分类示例

缺陷属性	场景 1	场景 2	场景 3	场景 4	场景 5
缺陷 ID	D080234	不适用	C080049	FM003	SD089
描述	在代码中未正确实现密码域	不适用	十六进制 HTTP 颜色代码为 000000，而不是 0000FF	指定要求中不正确的货币单位	根据网点环境管理规范的要求，设计未包括电池低电量警报

表B.2 (续)

缺陷属性	场景 1	场景 2	场景 3	场景 4	场景 5
状态	开	不适用	开	关	关
资产	网点低柜系统 服务器	不适用	人力资源管理	全部金融服务	网点业务连续性
工件	Login.asp	不适用	Cdisplay.c	核心银行需求 分析书	银行网点运行环 境系统设计任务 书
版本检测	V6.4	不适用	V3.4	初始文档	V1.0
版本修正		不适用	V4.0	V1.0	V2.0
优先级		不适用	低	高	高
严重性	关键	不适用	小	大	大
概率	高	不适用	高	低	高
影响	功能性	不适用	可用性	功能性	功能性
类型	接口	不适用	数据	数据	其他
模式	缺失	不适用	错误	错误	缺失
嵌入活动	编码	不适用	编码	需求	设计
检测活动	生产	不适用	客户测试	同行评审	生产
关联失效	F080001	不适用	FID001	不适用	DID005
关联变更	C080049	不适用	C_052	CHG_005	SWC_005
处置		不适用		改正	改正

参 考 文 献

- [1] IEEE std 1044TM—2009 Standard Classification for Software Anomalies
-