|  |  |
| --- | --- |
| ICS  | 91.120.25 |
| CCS  | L70 |

|  |
| --- |
|  21 |

辽宁省地方标准

DB21/T XXXX—XXXX

软件工程 软件测试缺陷等级判定方法

Method for determining defect level in software engineering software testing

（征求意见稿）

XXXX - XX - XX发布

XXXX - XX - XX实施

辽宁省市场监督管理局  发布

目次

[前言 II](#_Toc135138159)

[1 范围 3](#_Toc135138160)

[2 规范性引用文件 3](#_Toc135138161)

[3 术语和定义 3](#_Toc135138162)

[4 缺陷分类 4](#_Toc135138163)

[4.1 缺陷说明 4](#_Toc135138164)

[4.2 缺陷类型划分 4](#_Toc135138165)

[5 缺陷说明 6](#_Toc135138166)

[5.1 缺陷属性 7](#_Toc135138167)

[5.2 缺陷优先级 7](#_Toc135138168)

[5.3 缺陷状态 8](#_Toc135138169)

[5.4 缺陷来源 8](#_Toc135138170)

[5.5 缺陷原因 9](#_Toc135138171)

[5.6 缺陷修改次数 9](#_Toc135138172)

[5.7 缺陷生命周期 9](#_Toc135138173)

[5.8 缺陷严重等级 10](#_Toc135138174)

[6 判定方法 11](#_Toc135138175)

[6.1 严重级（P1） 11](#_Toc135138176)

[6.2 重要级（P2） 11](#_Toc135138177)

[6.3 中等级（P3） 12](#_Toc135138178)

[6.4 次要级（P4） 14](#_Toc135138179)

[6.5 有待改进（P5） 14](#_Toc135138180)

[附录A（资料性） 测评缺陷通报单模板 16](#_Toc135138181)

[附录B（资料性） 缺陷记录单模板 17](#_Toc135138182)

1. 前言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由辽宁省工业和信息化厅提出并归口。

本文件起草单位：北方实验室（沈阳）股份有限公司、辽宁鲲鹏生态创新中心有限公司、辽宁省标准化研究院。

本文件主要起草人：

本文件发布实施后，任何单位和个人如有问题和意见建议，均可以通过来电和来函等方式进行反馈，我们将及时答复并认真处理，根据实际情况依法进行评估及复审。

归口管理部门通讯地址：辽宁省工业和信息化厅（沈阳市皇姑区北陵大街45-2号），联系电话：024-86893258。

标准起草单位通讯地址：北方实验室（沈阳）股份有限公司（沈阳市浑南新区三义街6-1号21层），联系电话：024-83785841/83785849。

软件工程 软件测试缺陷等级判定方法

* 1. 范围

本标准规定了计算机软件生存周期内软件缺陷等级判定方法。

本标准适用于计算机软件生存周期全过程以及计算机软件的开发机构、测试机构相关人员。

* 1. 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；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 信息技术 软件工程术语

* 1. 术语和定义

GB/T 11457界定的以及下列术语和定义适用于本文件。

正确性 correctness

系统或部件在它的规格说明、设计和实现中无故障的程度；软件、文档或其他项满足规定的需求的程度；软件、文档或其他项满足用户需要和期望的程度（不管是否规定）。

崩溃 crash

计算机系统或部件的突然的和完全地失效。

周期 cycle

一种时间阶段，期间完成一组事件。

注：软件开发周期software development cycle和软件生存期software life cycle。

死锁 deadlock

由于两个或多个设备或进程，每一个等待赋给另一个的资源引起计算机处理停顿的情况。

缺陷故障 defect fault

硬件设备或部件中的缺陷以及在计算机程序中不正确的步骤、过程或数据定义。

注：此定义最初由容错系统使用。在通常用法中，术语“差错（error）”和“隐错（bug）”表示同样含义。

界面接口 interface

一个共享的边界及信息跨越边界传送。

注1：连接两个或多个其他部件，为了相互间传送信息的硬件或软件部件。

注2：连接两个或多个部件，为了在相互间传送信息。

模块 module

离散的程序单位或程序中一个能逻辑地分开的部分，且对于编译、与其他单位相结合和装入来说是可标识的。

注1：汇编程序、编译程序、连接编辑程序或执行的例行程序的输入或输出。

注2：术语“模块（module）”、“部件（component）”和“单位（unit）”常常是可互换的或以不同方法定义为另一个的子元素，取决于上下文。这些术语之间的关系尚未标准化。

参数 parameter

一种变量，针对一特定的应用程序可给定一固定值。或用于在软件模块之间传递值的常数、变量或表达式。

精度 precision

对于说明的量的精确或差异的程度，例如2位十进制数字对5位十进制数字。

过程 process

进程

为给定目的所执行的步骤序列，或由操作系统调度程序管理的一组可执行单元。

* 1. 缺陷分类
		1. 缺陷说明

软件缺陷指软件系统中存在的某种破坏正常运行能的问题、错误，或者隐藏的功能或性能上的缺陷。在软件中如果存在某种缺陷，将会导致产品不能满足某些用户功能。一般情况下，在软件系统中存在以下任意情况就被称之为软件缺陷：

1. 软件中未实现需求规格中标明需要实现的功能；
2. 软件中实现了需求规格中未标明的功能；
3. 软件中出现了需求规格中标明不能出现的功能；
4. 软件的功能超出了需求规格的范围；
5. 软件使用起来不方便或者难以理解、运行速度慢，或者最终用户认为不好用等。

对于一个软件缺陷的描述，需要包括缺陷的标识、缺陷的类型、缺陷的严重程度、缺陷的优先级、所处状态和缺陷的起源与来源等。

在判定过程中发现的缺陷，一般可分为如下几类，分别为：功能问题、接口问题、逻辑问题、计算问题、数据问题、用户界面问题、文档问题、性能问题、配置问题、标准问题、环境问题、兼容问题、其他问题等。

* + 1. 缺陷类型划分

缺陷类型划分见《缺陷种类表》（表1）。

1. 缺陷种类表

| **序号** | **缺陷类型** | **描述** | **子类型** |
| --- | --- | --- | --- |
| 1 | 功能问题F-Function | 影响了重要的特性、用户界面、产品接口、硬件结构接口和全局数据结构。并且设计文档需要正式的变更。如指针，循环，递归，功能等缺陷。 | 功能错误 |
| 功能缺失 |
| 功能超越 |
| 设计二义性 |
| 算法错误 |
| 2 | 接口问题I-Interface | 与其他组件、模块或设备驱动程序、调用参数、控制块或参数列表相互影响的缺陷。 | 模块间接口 |
| 模块内接口 |
| 公共数据使用 |
| 3 | 逻辑问题L-Logic | 需要进行逻辑分析，进行代码修改，如循环条件等。 | 分支不正确 |
| 重复的逻辑 |
| 忽略极端条件 |
| 不必要的功能 |
| 误解 |
| 条件判定错误 |
| 循环不正确 |
| 错误的变量检查 |
| 计算顺序错误 |
| 逻辑顺序错误 |
| 4 | 计算问题C-Computation | 等式、符号、操作符或操作书错误，精度不够、不适当的数据验证等缺陷。 | 等式错误 |
| 缺少运算符 |
| 错误的操作数 |
| 括号用法不正确 |
| 精度不够 |
| 舍入错误 |
| 符号错误 |

表1 缺陷种类表（续）

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **缺陷类型** | **描述** | **子类型** |
| 5 | 数据问题A-Assignment | 需要修改少量代码，如初始化或控制块。如声明、重复命名，范围、限定等缺陷。 | 初始化错误 |
| 存取错误 |
| 引用错误的变量 |
| 数组引用越界 |
| 不一致的子程序参数 |
| 数据单位不正确 |
| 数据维数不正确 |
| 变量类型不正确 |
| 数据范围不正确 |
| 操作符数据错误 |
| 变量定位错误 |
| 数据覆盖 |
| 外部数据错误 |
| 输出数据错误 |
| 输入数据错误 |
| 数据检验错误 |
| 6 | 用户界面问题U-User Interface | 人机交互特性：屏幕格式，确认用户输入，功能有效性，页面排版等方面的缺陷。 | 界面风格不统一 |
| 屏幕上的信息不可用 |
| 屏幕上的错误信息 |
| 界面功能布局和操作不合常规 |
| 7 | 文档问题D-Documentation | 影响发布和维护，包括注释等缺陷。 | 描述含糊 |
| 项描述不完整 |
| 项描述不正确 |
| 项缺少或多余 |
| 项不能验证 |
| 项不能完成 |
| 不符合标准 |
| 与需求不一致 |
| 文字排版错误 |
| 文档信息错误 |
| 注释缺陷 |

表1 缺陷种类表（续）

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **缺陷类型** | **描述** | **子类型** |
| 8 | 性能问题P-Performance | 不满足系统可测量的属性值，如：执行时间，事务处理速率等缺陷 |
| 9 | 配置问题B-Build/package/merge | 由于配置库、变更管理或版本控制引起的错误。 | 配置管理问题 |
| 编译打包缺陷 |
| 变更缺陷 |
| 纠错缺陷 |
| 10 | 标准问题N-Norms | 不符合各种标准的要求，如编码标准、设计符号等缺陷。 | 不符合编码标准 |
| 不符合软件标准 |
| 不符合行业标准 |
| 11 | 环境问题E-Environments | 由于设计、编译和运行环境引发的问题。 | 设计、编译环境 |
| 运行环境 |
| 12 | 兼容问题 | 软件之间不能正确地交互和共享信息。 | 操作平台不兼容 |
| 浏览器不兼容 |
| 分辨率不兼容 |
| 13 | 其他问题O-Others | 以上问题所不包含的其他问题。 |

* 1. 缺陷说明
		1. 缺陷属性

缺陷属性见《缺陷属性列表》（表2）。

1. 缺陷属性列表

| **序号** | **属性名称** | **说 明** |
| --- | --- | --- |
| 1 | 标识（Identifier） | 标记某个缺陷的唯一的符号，可以使用数字、字母组合来表示。 |
| 2 | 标题（Headline） | 对缺陷进行的简短描述。 |
| 3 | 描述（Description） | 对缺陷进行详细的描述，以便缺陷重现。 |
| 4 | 严重程度（Severity） | 指因缺陷引起的故障对软件产品的影响程度。 |
| 5 | 优先级（Priority） | 缺陷必须被修复的紧急程度。 |
| 6 | 状态（State） | 缺陷通过一个跟踪修复过程的进展情况。 |
| 7 | 来源（Source） | 指引起缺陷的起因。 |

* + 1. 缺陷优先级

缺陷优先级：指缺陷必须被修复的紧急程度。“优先级”的衡量抓住了在严重性中没有考虑的重要程度因素，见《缺陷优先级列表》（表3）。

1. 缺陷优先级列表

| **序号** | **缺陷优先级** | **优先级等级** | **描 述** |
| --- | --- | --- | --- |
| 1 | 立即解决（Resolve Immediately） | F1 | 导致判定无法继续进行，必须立刻进行修复；对用户产生很大影响，必须优先解决。 |
| 2 | 高度关注（Highly Focus） | F2 | 对此缺陷给以高度重视，应优先进行修复。 |
| 3 | 正常排队（Normal Queue） | F3 | 缺陷需要正常排队等待修复或列入软件发布清单。 |
| 4 | 低优先级（Not Urgent） | F4 | 缺陷可以在方便时被纠正。 |

* + 1. 缺陷状态

缺陷状态：指缺陷通知一个跟踪修复过程的进展情况，见《缺陷状态列表》（表4）。

1. 缺陷状态列表

| **序号** | **缺陷状态** | **描 述** |
| --- | --- | --- |
| 1 | 提交（Submitted） | 判定人员提交新的错误入库。 |
| 2 | 激活或打开（Active or Open） | 问题还没有解决，存在源代码中，确认“提交的缺陷”，等待处理。 |
| 3 | 拒绝（Rejected） | 拒绝“提交的缺陷”：不需要修复（Wont fix）或不是缺陷（Invalid）或缺陷已经被其他的软件判定人员发现（Duplicate）。 |
| 4 | 已修正或修复（Fixed or Resolved） | 已被开发人员检查、修复过的缺陷，通过单元判定，认为已解决但还没有被判定人员验证。 |
| 5 | 验证（Verify） | 缺陷验证通过。 |
| 6 | 关闭或非激活（Closed or Inactive） | 判定人员验证后，确认缺陷不存在之后的状态。 |
| 7 | 重新打开（Reopen） | 判定人员验证后，还依然存在的缺陷，等待开发人员进一步修复。 |
| 8 | 推迟（Deferred） | 这个软件缺陷在下一个版本中解决。 |
| 9 | 保留（On hold） | 由于技术原因或第三者软件的缺陷，开发人员暂不能修复的缺陷。 |
| 10 | 不能重现（Cannot duplicate） | 开发不能复现这个软件缺陷，需要判定人员检查缺陷复现的步骤。 |

* + 1. 缺陷来源

缺陷来源：缺陷引起的故障或事件第一次被检测到的阶段，见《缺陷来源列表》（表5）。

1. 缺陷来源列表

| **序号** | **缺陷原因** | **描 述** |
| --- | --- | --- |
| 1 | 需求（Requirement） | 由于需求的问题引起的缺陷 |
| 2 | 架构（Architecture） | 由于构架的问题引起的缺陷 |
| 3 | 设计（Design） | 由于设计的问题引起的缺陷 |
| 4 | 编码（Coding） | 由于编码的问题引起的缺陷 |
| 5 | 判定（Test） | 由于判定的问题引起的缺陷 |
| 6 | 集成（Integration） | 由于集成的问题引起的缺陷 |
| 7 | 用户（Customer） | 在用户使用阶段发现的缺陷 |
| 8 | 其他（Other） | 由于其他问题引起的缺陷 |

* + 1. 缺陷原因

缺陷原因：造成上述错误的根本因素，以寻求开发、判定人员可改进的地方，见《缺陷原因列表》（表6）。

1. 缺陷原因列表

| **序号** | **缺陷原因** | **描 述** |
| --- | --- | --- |
| 1 | 低级错误 | 正常的状态下不可能犯的错误。 |
| 2 | 疏忽 | 因马虎大意导致的缺陷。 |
| 3 | 考虑欠周全 | 因考虑不周全导致的缺陷。 |
| 4 | 理解不到位 | 对业务、需求、设计等理解不到位导致的缺陷。 |
| 5 | 其他 |  |

* + 1. 缺陷修改次数

缺陷修改次数：同一个缺陷被重新修复的次数，见《缺陷修改次数表》（表7）。

1. 缺陷修改次数表

| **序号** | **缺陷原因** | **描 述** |
| --- | --- | --- |
| 1 | 首次 | 同一缺陷只修复一次就能关闭。 |
| 2 | 两次 | 同一缺陷修复后，验证不通过，需重新打开，经过两次修复后关闭。 |
| 3 | 三次 | 同一缺陷修复后，验证不通过，需重新打开，经过三次修复后关闭。 |
| 4 | 多次 | 同一缺陷修复后，验证不通过，需重新打开，经过三次以上修复才关闭。 |

* + 1. 缺陷生命周期

生命周期的概念是一个物种从诞生到消亡经历了不同的生命阶段，那么，在判定过程中，软件缺陷生命周期并非判定人员提出后，就一直处于激活状态，应该指的是一个软件缺陷被发现、报告到这个缺陷被修复、验证直至最后关闭的完整过程。在整个软件缺陷生命周期中，通常是以改变软件缺陷的状态来体现不同的生命阶段。因此，对于一个软件判定人员来讲，需要关注软件缺陷在生命周期中的状态的变化，来跟踪项目进度和软件质量。一个简单、优化的软件缺陷生命周期状态，分别为：

1. New：新建，判定人员初始提单；
2. Assign：指派，产品负责人指派给相应开发人员；
3. Verified：确认，开发修复后产品负责人进行确认；
4. Open：打开，开发人员已修复，产品负责人未确认；
5. Duplicate：重复，开发人员确认已存在重复 BUG；
6. Deferred：延期，个别异常缺陷经产品负责人确认延期处理；
7. Reopened：激活，判定人员进行回归验证未通过缺陷；
8. Rejected：拒绝，开发人员经确认非缺陷；
9. Closed：关闭，判定人员经回归通过后关闭。

在实际的工作过程中，软件缺陷的生命周期基本上是不可能为：新建→指派→确认→关闭的简单流程，因此，需要考虑到其他各种可能发生的情况，现整理出如（图1）所示的一个较为复杂的软件缺陷生命周期流转图。

软件缺陷在生命周期中经历了数次的审阅和状态变化，最终，判定人员会根据回归判定结果来关闭软件缺陷来结束软件缺陷的生命周期。软件缺陷生命周期中的不同阶段是判定人员、开发人员和负责人员一起参与、协同判定的过程。软件缺陷一旦发现，便进入判定人员、开发人员、负责人员的严密监控之中，直至软件缺陷生命周期终结，这样即可保证在较短的时间内高效率地关闭所有的缺陷，缩短软件判定的进程，提高软件质量，同时减少开发、判定和维护成本。



1. 流程图
	* 1. 缺陷严重等级

在判定过程中，除了进行缺陷分类，同时，还需要对缺陷进行严重等级选择，其严重等级常规有如下几种，见《缺陷严重程度列表》（表8）。

1. 缺陷严重程度列表

| **序号** | **严重级别** | **缺陷严重等级** |
| --- | --- | --- |
| 1 | 严重（Critical） | P1 |
| 2 | 重要（Major） | P2 |
| 3 | 中等（Normal） | P3 |
| 4 | 次要（Minor） | P4 |
| 5 | 有待改进（Enhancement） | P5 |

* 1. 判定方法
		1. 严重级（P1）
			1. 严重级（P1）缺陷描述

不能执行正常工作功能或实现重要功能，包括：

1. 可能有灾难性的后果，如造成系统崩溃，造成事故等；
2. 数据库错误，如数据丢失等。
	* + 1. 严重级（P1）缺陷判定方法

被测软件存在以下问题判定为P1级别缺陷：

1. 系统崩溃；
2. 数据丢失；
3. 数据库报错；
4. 功能错误；
5. 功能缺失；
6. 功能超越；
7. 设计二义性；
8. 算法错误；
9. 模块间接口；
10. 模块内接口；
11. 公共数据使用；
12. 由于程序所引起的死机，非法退出；
13. 程序死循环；
14. 性能与需求不一致（压力判定）；
15. 存在安全性与保密性问题；
16. 文件打开与保存错误。
	* 1. 重要级（P2）
			1. 重要级（P2）缺陷描述

产生错误的结果，导致系统不稳定，运行时好时坏，严重地影响系统要求或基本功能实现的问题，如：

1. 造成数据库不稳定的错误；
2. 在说明中的需求未在最终系统中实现；
3. 程序无法运行，系统意外退出；
4. 业务流程不正确。
	* + 1. 重要级（P2）缺陷判定方法

被测软件存在以下问题判定为P2级别缺陷：

1. 系统不稳定；
2. 业务逻辑错误；
3. 无法正常运行；
4. 系统未实现需规功能；
5. 分支不正确；
6. 重复的逻辑；
7. 忽略极端条件；
8. 不必要的功能；
9. 误解；
10. 条件判定错误；
11. 循环不正确；
12. 错误的变量检查；
13. 计算顺序错误；
14. 逻辑顺序错误；
15. 等式错误；
16. 缺少运算符；
17. 错误的操作数；
18. 括号用法不正确；
19. 精度不够；
20. 舍入错误；
21. 符号错误；
22. 数据计算错误；
23. 数据约束错误；
24. 数据输入、输出错误；
25. 数据库发生死锁；
26. 数据库的表、业务规则、缺省值未加完整性等约束条件；
27. 数据库连接错误；
28. 数据通讯错误；
29. 程序接口错误；
30. 硬件接口、通讯错误；
31. 程序功能无法实现；
32. 程序功能实现错误。
	* 1. 中等级（P3）
			1. 中等级（P3）缺陷描述

不正确的，但不会影响系统稳定性的：

1. 过程调用或其它脚本错误；
2. 系统刷新错误；
3. 产生错误结果，如计算结果错误，数据不一致等；
4. 功能的实现有问题，如在系统实现的界面上，一些可接受输入的控件点击后无作用，对数据库的操作不能正确实现；
5. 编码时数据类型、长度定义错误；
6. 虽然正确性，功能不受影响，但系统性能和响应时间受到影响；
7. 对于输入数据没有进行必要的类型校验。
	* + 1. 中等级（P3）缺陷判定方法

被测软件存在以下问题判定为P3级别缺陷：

1. 调用异常；
2. 刷新错误；
3. 数据计算错误；
4. 功能无用；
5. 长定义错误；
6. 数据类型错误；
7. 响应时间过慢；
8. 输入项未校验；
9. 初始化错误；
10. 存取错误；
11. 引用错误的变量；
12. 数组引用越界；
13. 不一致的子程序参数；
14. 数据单位不正确；
15. 数据维数不正确；
16. 变量类型不正确；
17. 数据范围不正确；
18. 操作符数据错误；
19. 变量定位错误；
20. 数据覆盖；
21. 外部数据错误；
22. 输出数据错误；
23. 输入数据错误；
24. 数据检验错误；
25. 操作平台不兼容；
26. 浏览器不兼容；
27. 分辨率不兼容；
28. 程序功能无法实现；
29. 程序功能实现错误；
30. 脚本错误；
31. 软件产品的编译，打包，安装，卸载错误；
32. 性能不满足系统可测量的属性值，如：执行时间，事务处理速率等缺陷。
	* 1. 次要级（P4）
			1. 次要级（P4）缺陷描述

不正确的，但有使系统使用起来不太方便的错误，重点指系统的 UI 问题：

1. 系统的提示语不明确，不简明；
2. 滚动条无效；
3. 可编辑区和不可编辑区不明显；
4. 光标跳转设置不好，鼠标（光标）定位错误；
5. 上下翻页，首尾页定位错误；
6. 界面不一致，或界面不正确；
7. 日期或时间初始值错误（起止日期、时间没有限定）。
	* + 1. 次要级（P4）缺陷判定方法

被测软件存在以下问题判定为P4级别缺陷：

1. 系统提示不明确；
2. 滚动条失效；
3. 编辑区不明显；
4. 定位错误；
5. 日期初始化错误；
6. 配置管理问题；
7. 编译打包缺陷；
8. 变更缺陷；
9. 纠错缺陷；
10. 不符合编码标准；
11. 不符合软件标准；
12. 不符合行业标准；
13. 设计、编译环境；
14. 使操作者不方便或者遇到麻烦，但不影响执行工作功能的实现；
15. 界面、控件的摆布、图标、输入输出不规范；
16. 删除操作未给出提示；
17. 长时间操作未给出提示；
18. 提示窗口文字未采用行业术语；
19. 出错没有提示；
20. 不符合编码标准；
21. 辅助说明描述不清楚、不规范；
22. 快捷键无效，快捷键错误操作；
23. 打印内容、格式错误。
	* 1. 有待改进（P5）
			1. 有待改进（P5）缺陷描述

系统中值得改良的问题：

1. 容易给用户误解和歧义的提示；
2. 界面需要改进的，某个控件没有对齐等；
3. 对有疑虑的部分，提出修改建议。
	* + 1. 有待改进（P5）缺陷判定方法

被测软件存在以下问题判定为P5级别缺陷：

1. 界面风格不统一；
2. 友好提示错误；
3. 控件未对齐；
4. 屏幕上的信息不可用；
5. 屏幕上的错误信息；
6. 界面功能布局和操作不合常规；
7. 描述含糊；
8. 项描述不完整；
9. 项描述不正确；
10. 项缺少或多余；
11. 项不能验证；
12. 项不能完成；
13. 不符合标准；
14. 与需求不一致；
15. 文字排版错误；
16. 文档信息错误；
17. 注释缺陷。

1. （资料性）
缺陷记录单模板
2. （资料性）
测评缺陷通报单模板

测评缺陷通报单模板见表A.1。

* 1. 测评缺陷通报单

|  |  |  |  |
| --- | --- | --- | --- |
| 软件名称: |  | 版本号: |  |
| 委托单位: |  | 项目编号： |  |
| 测评单位: |  | 文档编号: |  |
| 测评人员： |  |
| 测评时间： |  测评地点：  |
| **缺陷编号** | **缺陷级别** | **缺陷描述** | **备注** |
| WT001 |  |  |  |
| WT002 |  |  |  |
| WT003 |  |  |  |

1. （资料性）
缺陷记录单模板

缺陷记录单模板见表B.1。

* 1. 缺陷记录单

|  |  |  |  |
| --- | --- | --- | --- |
| 软件名称 |  | 缺陷编号 |  |
| 文档编号 |  | 版本号 |  |
| 测试人员 |  |
| 测试日期 |  |
| 硬件平台 |  |
| 软件系统 |  |
| 缺陷级别 |  |
| 缺陷优先级 |  |
| 缺陷概述 |  |
| 缺陷详述 |  |

