ICS 35.080
L 77

中华人民共和国国家标准

GB/T 37970—2019

软件过程及制品可信度评估

Trustworthiness assessment for software process and artifact

2019-08-30 发布　　2020-03-01 实施

国家市场监督管理总局
中国国家标准化管理委员会　发布

目　　次

前　言

本标准按照 GB/T 1.1—2009 给出的规则起草。

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这些专利的责任。

本标准由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本标准起草单位:中国科学院软件研究所、中国电子技术标准化研究院、南京大学、中国科学院科技战略咨询研究院、厦门理工学院、上海计算机软件技术开发中心、重庆邮电大学、北京拓尔思信息技术股份有限公司、罗普特(厦门)科技集团有限公司、北京知道未来信息技术有限公司。

本标准主要起草人:王丹丹、王德鑫、刘增志、赖宜亮、韦庆杰、吴登生、胡芸、冯志超、贺劼、王林章、崔建峰、张旸旸、王青。

引　言

质量形成于过程。软件可信的基本问题是需要有证据证明软件产品能够满足用户的需求，而这些证据散布于软件开发的整个生存周期。所以，软件是否可信不能仅依赖于对最终产品的测试，对软件可信的评估和确认需要软件开发过程中各类相关证据的支持。本标准旨在从软件过程及制品的角度，建立系统化的可信度评估模型，指导软件开发在过程中采集合适的数据，以形成证据，支持证据驱动下的软件可信度评估。

软件可信并非一个新的质量特性，而是对软件的功能性、安全性、可维护性、可靠性等各种质量特性满足需求的程度的测量。受传统行业全面质量管理理论的启发，本标准提出面向过程的方法，通过关注和提高软件开发过程的质量，来提高软件的可信度。CMMI 是被业界广泛采用的软件过程管理框架，目前运行的 CMMI V1.3 包括 22 个过程域，用软件开发过程的知识技术解决软件管理流程，强调改进软件过程能力，提高软件过程的成熟度，从而帮助企业有预期地、稳定地在预算内按时交付满足用户要求的产品。

本着兼容国际流行技术的准则，提出软件过程可信原则覆盖 CMMI V1.3 的 22 个过程域，另外扩充了对可信实体的保障，并通过制品可信原则对软件过程的制品进行了可信增强。形成的软件过程及制品可信证据体系，将软件质量分解到软件开发的各个阶段和过程，有目标地采集过程数据，形成覆盖软件开发全过程的证据链，并最终对交付的软件产品满足预期质量目标的可信度进行评估。

软件过程及制品可信度评估

1 范围

本标准规定了软件过程和制品可信度评估使用的模型、可信等级和评估方法。

本标准适用于指导开发软件产品的组织建立可信的软件过程，并采集数据，积累证据以支持对软件产品质量的信心。同时也适用于第三方评估认证机构建立评估方法，对目标软件产品及其开发组织开展第三方评估。

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 19001—2016 质量管理体系 要求

GB/T 25000.10—2016 系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第10部分：系统与软件质量模型

GB/T 25000.23—2019 系统和软件工程 系统与软件质量要求和评价(SQuaRE) 第23部分：系统与软件产品质量测量

3 术语和定义、缩略语

3.1 术语和定义

GB/T 25000.10—2016、GB/T 25000.23—2019 和 GB/T 19001—2016 界定的以及下列术语和定义适用于本文件。

3.1.1

过程 process

为给定目的所执行的步骤序列，例如，软件开发过程。

[GB/T 11457—2006，定义 2.1183]

3.1.2

软件过程 software process

由组织或项目使用的，用以计划、管理、执行、监控和改进其软件相关活动的过程或过程的集合。

[GB/T 11457—2006，定义 2.1512]

3.1.3

软件开发全过程 whole process of software development

软件开发中的需求、设计、编码和测试过程。

注：与 GB/T 11457—2006 中定义的开发过程相比，为便于可信指标的分类，软件开发全过程专指需求、设计、编码和测试四个过程。

3.1.4

制品 artifact

由某一种软件开发过程所使用的或产生的一种信息的物理件。制品的实例有模型、源文件、文字和

二进制可执行文件。制品可构成可部署构件的实现。

[GB/T 11457—2006,定义 2.76]

3.1.5

过程可信度 process trustworthiness

软件过程能够生产满足期望产品的信心度。

3.1.6

制品可信度 artifact trustworthiness

软件过程产生的制品能够满足期望产品质量的信心度。

3.1.7

实体 entity

通过测量其属性表述其特性的对象。

例如,一个对象可能是过程、产品、项目或资源。

[GB/T 20917—2007,定义 3.9]

3.1.8

可信原则 trustworthiness principle

由一组过程活动产生的证据来支持的,用于保障可信的通用要求。

3.1.9

可信过程域 trustworthy process area

以可信保障目标为导向的一类可信原则。

3.1.10

证据 evidence

针对软件开发活动中留下的数据(如文档、各种形式的记录、证言等),进行计算得到的,用于表明活动的表现是否满足要求的测量数据。

3.1.11

过程证据 process evidence

表征过程是否满足期望的过程可信度的证据。

注:该证据的数据类型分为布尔型、数值型、百分比型和等级型。

3.1.12

制品证据 artifact evidence

表征制品是否满足期望的制品可信度的证据。

注:该证据的数据类型分为布尔型、数值型、百分比型和等级型。

3.1.13

软件质量特性 software quality characteristic

支撑软件质量的软件质量属性的类别。

[GB/T 25000.10—2016,定义 3.23]

3.1.14

环境 environment

支撑软件开发的计算机、系统软件、开发工具、网络、基础架构、数据管理设施等软硬件条件或要求。

3.1.15

依从性 compliance

产品或系统遵循相关标准、约定或法规以及类似规定的程度。

3.1.16

形式化　formalization

描述系统性质的基于数学的技术。

3.1.17

规约　specification

为满足某种需求而提供的解决方案。

注：可称为规格说明。

3.2　缩略语

下列缩略语适用于本文件。

CMMI：能力成熟度集成模型(Capability Maturity Model Integration)

PA：过程域(Process Area)

4　软件可信度模型

4.1　模型概述

可信的基本要素包括可信的实体、可信的行为和可信的结果三个维度。

可信保障目标为：实体可信、行为可信和结果可信。实体可信分解为开发活动中涉及的人、交互通道、环境以及使用的方法和工具；行为可信分解为开发过程中的行为管理、制品管理和文档支持；结果可信分解为对过程产品的验证和确认，其中“验证”关注合适的过程制品检验方法，“确认”关注过程制品的质量满足需求的程度。

以可信保障目标为导向确定了七类可信原则，即 7 个可信过程域。其中实体安全、开发支持、文档化、可管理、可追溯、可验证关注过程的可信，定义为过程可信过程域，其对应的可信原则定义为软件过程可信原则，共包含 36 个软件过程可信原则；制品可信关注过程制品的可信，定义为制品可信过程域，其对应的可信原则定义为软件制品可信原则，共包含 1 个可信原则。可信原则与 CMMI 过程域的对比参见附录 A。

每类可信过程域包含若干可信原则，每个可信原则由面向不同开发阶段或软件开发全过程的一组活动来保障，活动产生的数据形成证据来支持可信原则实现程度的评估。

软件可信度模型见图 1。

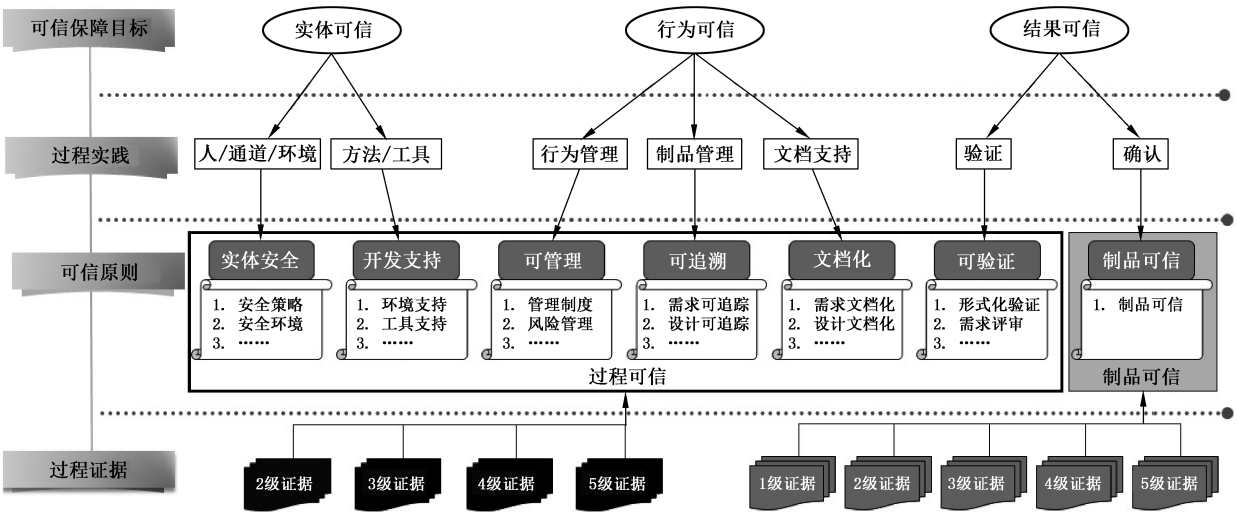

图 1　软件可信度模型

可信原则对应的证据结构见表1。其中证据等级表示该证据从该级开始要求，其满足程度可不断提高，最高等级大于或等于证据等级。

注：一级可信等级表示有关的活动已执行，但过程处于无定义和管理的状态下，对活动的可信没有要求，因此本模型没有一级过程证据。

表1 可信原则的证据结构

证据类型	证据
二级证据	证据2.1、证据2.2……
三级证据	证据3.1、证据3.2……
四级证据	证据4.1、证据4.2……
五级证据	证据5.1、证据5.2……

4.2 软件过程可信原则

4.2.1 概述

软件过程可信原则共有36个，按照可信过程域类别和软件开发阶段两个维度的分类组织见表2。横向维度表示软件过程可信原则涉及的软件开发阶段，包括需求、设计、编码、测试等必要的典型阶段，适合包括瀑布模型、敏捷开发在内的所有开发形式。各阶段的具体内容如下：

a) 环境：指支撑软件开发的计算机、系统软件、开发工具、网络、基础架构、数据管理设施等软硬件条件或要求，共8个可信原则；

b) 需求：软件开发的需求阶段，包括5个可信原则；

c) 设计：软件开发的设计阶段，包括5个可信原则；

d) 编码：软件开发的编码阶段，包括5个可信原则；

e) 测试：软件开发的测试阶段，包括4个可信原则；

f) 软件开发全过程：软件开发的需求、设计、编码和测试阶段，这部分的可信原则贯彻软件开发全过程，共9个可信原则。

表2 软件过程相关的可信原则分类组织视图

<table>
<tr><th rowspan="2">过程域</th><th colspan="6">阶段</th><th rowspan="2">合计</th></tr>
<tr><th>环境</th><th>需求</th><th>设计</th><th>编码</th><th>测试</th><th>软件开发全过程</th></tr>
<tr><td rowspan="4">实体安全</td><td rowspan="4"></td><td rowspan="4" colspan="4"></td><td>安全政策</td><td rowspan="4">4</td></tr>
<tr><td>安全管理策略</td></tr>
<tr><td>安全环境</td></tr>
<tr><td>信息安全工具支持</td></tr>
<tr><td rowspan="4">开发支持</td><td>开发环境工具</td><td rowspan="4">需求分析工具</td><td rowspan="4">设计工具</td><td rowspan="4">源代码分析工具</td><td rowspan="4">测试工具</td><td rowspan="4"></td><td rowspan="4">8</td></tr>
<tr><td>重用支持</td></tr>
<tr><td>开源支持</td></tr>
<tr><td>采购支持</td></tr>
</table>

表 2（续）

<table>
<tr><td rowspan="2">过程域</td><td colspan="6">阶段</td><td rowspan="2">合计</td></tr>
<tr><td>环境</td><td>需求</td><td>设计</td><td>编码</td><td>测试</td><td>软件开发全过程</td></tr>
<tr><td rowspan="5">可管理</td><td>知识共享</td><td rowspan="5" colspan="4"></td><td>风险管理</td><td rowspan="5">8</td></tr>
<tr><td>管理制度化</td><td>计划</td></tr>
<tr><td rowspan="3">环境完整性</td><td>测量与分析</td></tr>
<tr><td>配置管理</td></tr>
<tr><td>过程审计</td></tr>
<tr><td>文档化</td><td></td><td>需求分析文档化</td><td>设计文档化</td><td>源码文档化</td><td>测试文档化</td><td></td><td>4</td></tr>
<tr><td rowspan="2">可验证</td><td rowspan="2">形式化验证
要求</td><td>形式化需求验证</td><td>形式化设计
验证</td><td>形式化代码
验证</td><td rowspan="2">测试评审</td><td rowspan="2"></td><td rowspan="2">8</td></tr>
<tr><td>需求分析评审</td><td>设计评审</td><td>源代码评审</td></tr>
<tr><td>可追溯</td><td></td><td>需求
可追踪性</td><td>设计
可追踪性</td><td>源代码
可追踪性</td><td>测试
可追踪性</td><td></td><td>4</td></tr>
<tr><td>合计</td><td>8</td><td>5</td><td>5</td><td>5</td><td>4</td><td>9</td><td>36</td></tr>
</table>

4.2.2 实体安全过程域的可信原则

实体安全过程域共有 4 个可信原则，包括：

a） 安全政策：所有的软件开发者执行开发活动应遵守明确定义的安全政策；

b） 安全管理策略：生存周期活动的执行需要由至少两个有资格的开发人员的认同和参与；

c） 安全环境：应明确安全机制，保证全生存周期活动不会受到未授权方法影响；

d） 信息安全工具支持：根据清晰定义的安全策略，所有确定的软件生存周期活动应被软件工程环境自动控制。

4.2.3 开发支持过程域的可信原则

开发支持过程域共有 8 个可信原则，包括：

a） 开发环境工具：软件工程环境和所有的软件工具应根据一个明确的选择策略来进行选择，选择策略中应考虑可信等级、成熟度、文档和源码的可获取性等因素；

b） 重用支持：是否要构建可重用的组件以及可重用的程度；

c） 开源支持：开源软件应接受选择，清晰的选择政策应考虑可信等级、成熟度、文档、可获取的源码和软件许可协议；

d） 采购支持：应与供应商建立采购合同以及评价采购产品质量的等级；

e） 需求分析工具：应使用需求分析工具以支持需求规约、一致性检查和文档生成；

f） 设计工具：在设计中应采用设计工具以维护设计/需求的跟踪映射关系并生成设计文档；

g） 源代码分析工具：应使用测量复杂度和风格的源代码分析工具和步骤来分析所有开发的代码；

h） 测试工具：软件工程环境应包含一个创造、执行、文档化和分析测试完整的测试工具集。

4.2.4 可管理过程域的可信原则

可管理过程域共有8个可信原则，包括：

a) 知识共享：每个软件开发活动的组件，包括需求、源码、设计、测试、软件工具、方法和支撑活动等都应与至少两个人员相关，这些人员非常熟悉这些组件的细节、隐含的意义和所考虑的选择方案；

b) 管理制度化：根据管理文档，应由有资格的人员对软件工程环境、软件工具和开发的软件进行维护；

c) 环境完整性：对于识别软件工程环境组件的变更应有一个明确的步骤，如果有需要，恢复环境的完整性；

d) 风险管理：与软件开发活动相关的风险都应被明确地识别，风险移除策略应被文档化；

e) 计划：对于所有软件开发活动的详细计划应在软件开发计划书中描述，软件开发的管理也应遵循计划书中所描述的方法；

f) 测量与分析：应对软件过程和制品进行合适的测量和分析，以准确地理解过程和制品的状态，及时了解过程的偏差，采取合理的纠正措施；

g) 配置管理：应建立一个配置管理系统，包括关于配置项识别、审核、控制和审计的明确机制和步骤。所有的配置项应保存在存放处以维护软件版本、软件修改请求和变更；

h) 过程审计：在全生存周期中负责软件开发过程的总体和阶段计划、任务分配、沟通、协调、跟踪、控制，保证任务质量和进度，并根据任务执行情况追溯到每一个开发环节。

4.2.5 文档化过程域的可信原则

文档化过程域共有4个可信原则，包括：

a) 需求分析文档化：除了软件需求规约和接口规约，所有帮助理解需求分析过程、重要的需求分析决策的原理等有用信息都应被文档化；

b) 设计文档化：除了软件设计说明书和接口设计说明书外，设计活动的特性、考虑到的重要的设计选择项和重要的设计理由都应被文档化；

c) 源代码文档化：源码和软件编码活动的特征应被文档化；

d) 测试文档化：除了软件测试计划书、软件测试描述书和软件测试报告以外，软件组件和配置项测试活动的特征也应被文档化。

4.2.6 可验证过程域的可信原则

可验证过程域共有8个可信原则，包括：

a) 形式化验证要求：按照形式化验证要求，所有的形式化规约和验证活动都应遵循一个合适的方法，包括使用形式化规约和验证工具、文档和同行评审等；

b) 形式化需求验证：除了非形式化的需求规约以外，需求文档应用一个形式化的框架来规约；

c) 需求分析评审：应由一个同行评审组对需求分析进行同行评审以保证软件需求分析的完整性、一致性和正确性；

d) 形式化设计验证：应对形式化的设计规约进行形式化设计验证以确保满足其需求；

e) 设计评审：应由一个同行评审组对设计进行同行评审以保证软件设计的完整性、一致性和正确性；

f) 形式化代码验证：形式化验证代码是否满足形式化规约，所有的形式化规约和验证活动都应遵

循一个包含了使用形式化规约和验证工具、文档、同行评审和可跟踪映射的方法；

g) 源代码评审：应由一个同行评审组对源码进行同行评审以保证软件源码和计算机软件单元测试的完整性、一致性和正确性；

h) 测试评审：应由一个同行评审组对测试进行同行评审以保证软件测试的完整性、一致性和正确性。

4.2.7 可追溯过程域的可信原则

可追溯域共有 4 个可信原则，包括：

a) 需求可跟踪性：对于明确的系统需求或客户来源，所有的软件需求应保持可跟踪性；

b) 设计可跟踪性：设计的各方面和需求应是互相可跟踪的；

c) 源代码可跟踪性：所有的源码对于设计和计算机软件单元测试应是可跟踪的，设计对于源码也应如此；

d) 测试可跟踪性：所用软件组件和配置项的测试对于需求应是可跟踪的，源码和需求对于组件和配置项的测试也应如此。

4.3 软件制品可信原则

软件制品可信原则只有一项。按照软件质量特性和软件生存周期阶段两个维度，软件制品可信原则对应证据的分类组织视图见表 3。横向维度表示制品可信原则涉及的软件全生存周期，包括设计、编码、测试、交付、维护和软件开发全过程。纵向维度表示 GB/T 25000.10—2016 定义的 8 个软件质量特性：功能性、性能效率、兼容性、易用性、可靠性、信息安全性、维护性和可移植性。

表 3 软件制品可信原则对应证据的分类组织视图

可信过程域	可信原则	质量特性	阶段						合计
			设计	编码	测试	交付	维护	软件开发全过程	
制品可信	制品可信	功能性	2	2	6	2	0	1	13
		性能效率	0	0	0	11	0	1	12
		兼容性	2	0	2	0	0	1	5
		易用性	1	0	3	7	0	1	12
		可靠性	3	0	1	2	4	2	12
		信息安全性	5	0	2	4	0	1	12
		维护性	1	0	9	1	0	1	12
		可移植性	0	0	0	5	0	1	6
合计		8	14	2	23	32	4	9	84

4.4 软件过程可信证据

4.4.1 概述

由于不同软件系统的可信要求不同，并不要求所有证据都要采集和测量，不同级别的证据支持不同级别的可信要求。例如，当软件的可信要求为三级时，只需要评估各个原则三级及三级以下的证据。软件可信度模型一共定义了 133 个过程证据，按照开发阶段和可信过程域所对应的两种等级分布，分别见

表 4 和表 5。具体证据见附录 B。

表 4　按照软件开发阶段表示过程证据的等级分布

开发阶段	可信原则	证据等级					
		一级	二级	三级	四级	五级	证据数总计
环境	8	0	21	9	0	0	30
需求	5	0	12	0	1	0	13
设计	5	0	10	0	2	0	12
编码	5	0	12	0	0	2	14
测试	4	0	17	4	0	0	21
软件开发全过程	9	0	31	9	2	1	43
总计	36	0	103	22	5	3	133

表 5　按照可信过程域表示过程证据的等级分布

可信过程域	可信原则	证据等级					
		一级	二级	三级	四级	五级	总计
实体安全	4	0	10	4	0	0	14
开发支持	8	0	25	8	0	0	33
可管理	8	0	28	10	2	1	41
文档化	4	0	11	0	0	0	11
可验证	8	0	20	0	3	1	24
可追溯	4	0	9	0	0	1	10
证据数总计	36	0	103	22	5	3	133

以下将从 6 个可信过程域的角度，提出软件过程可信原则的证据要求。

4.4.2　实体安全过程域的可信证据

4.4.2.1　概述

实体安全过程域共有 4 个可信原则，涉及软件开发全过程，共包含 14 个可信证据。

4.4.2.2　软件开发全过程

软件开发全过程可信原则下的可信证据如下：

a）　安全政策：安全政策的建立、安全相关政策的了解、培训的开展；

b）　安全管理策略：背景调查、人员对可信知识了解程度、信息安全管理的程度、项目遵循安全要求的程度、安全管理相关培训的开展、共享监控机制；

c）　安全环境：安全工作环境标准的建立、软件过程和资产访问权限和环境的建立、网络服务安全性；

d）　信息安全工具支持：自动化工具对权限控制的支持程度、组织结构对权限支持。

4.4.3 开发支持过程域的可信证据

4.4.3.1 概述

开发支持过程域共有8个可信原则，涉及环境、需求、设计、编码、测试五个阶段，共包含33个可信证据。

4.4.3.2 环境阶段

环境阶段可信原则下的可信证据如下：

a) 开发环境工具：开发工具支持、工作环境标准的建立、环境与工具相关培训的开展；
b) 重用支持：重用准则的建立、重用分析、支持和维护重用软件架构的开发、重用部件的质量评价、可重用部件设计标准的建立；
c) 开源支持：对选择的开源软件的调研、对采用开源软件的成本估算、开源软件提供方持续的技术支持、开源软件的质量评价、采用开源软件的风险评估；
d) 采购支持：供应商目录的建立、采购合同的建立和维护、采购产品质量的评价。

4.4.3.3 需求阶段

需求阶段的可信原则下可信证据主要为需求分析工具，即人员对需求分析工具的掌握程度、软件需求分析自动化环境、自动化工具对需求分析的支持程度。

4.4.3.4 设计阶段

设计阶段的可信原则下可信证据主要为设计工具，即人员对设计工具的掌握程度、设计自动化环境。

4.4.3.5 编码阶段

编码阶段的可信原则下可信证据主要为源代码分析工具，即人员对代码分析工具的掌握程度、自动化工具对源代码分析的支持程度、源代码库。

4.4.3.6 测试阶段

测试阶段的可信原则下可信证据主要为测试工具，即测试工具类别的完整性、测试环境的搭建、人员对测试工具的掌握程度、测试数据和测试用例的设计、自动化工具对软件测试的支持程度、单元测试策略、集成测试策略、配置项测试策略、组件测试策略。

4.4.4 可管理过程域的可信证据

4.4.4.1 概述

可管理过程域共有8个可信原则，涉及环境、软件开发全过程，共包含41个可信证据。

4.4.4.2 环境阶段

环境阶段可信原则下的可信证据如下：

a) 知识共享：支持知识共享的组织资产库覆盖的范围、共享知识的获取方式、组织资产的改进与维护、组织培训能力建立与维护；
b) 管理制度化：管理规程的建立、职责和权限的明确、培训的开展、管理规程的覆盖范围、决策过程管理规范的建立；

c) 环境完整性：软件工程环境变更权限的配置、软件工程环境变更的记录、软件工程环境的变更审查。

4.4.4.3 软件开发全过程

软件开发全过程可信原则下的可信证据如下：

a) 风险管理：风险管理计划、发现识别和评价标准和指南的建立、风险管理策略的建立、项目风险管理的效果；
b) 计划：软件开发计划书的编制、项目的人员配置和利益相关方的明确、项目过程制品和里程碑的明确定义和配置管理、风险的识别和评估、项目计划的成熟程度、项目利用与贡献组织资产的程度；
c) 测量与分析：测量目标的建立、相关的测量元和数据采集及测量和存储方法的建立、测量体系覆盖的程度、项目开展测量分析的程度、测量分析结果在组织级别的保存和分析、测量数据的采集、分析和存储的工具支持、选择的关键过程控制属性在统计意义上的情况、过程异常的分析和改进、过程和产品持续改进机会的量化分析；
d) 配置管理：配置管理策略、配置基线管理、配置变更管理、配置状态审计、软件配置管理工具的使用、配置管理人员的成熟度；
e) 过程审计：内控制度的有效性、审计方法的充分性、审计过程的规范性、审计数据。

4.4.5 文档化过程域的可信证据

4.4.5.1 概述

文档化过程域共有 4 个可信原则，涉及需求、设计、编码、和测试四个开发阶段，共包含 11 个可信证据。

4.4.5.2 需求阶段

需求阶段的可信原则下的可信证据主要为需求分析文档化，即系统需求文档化、用户需求文档化。

4.4.5.3 设计阶段

设计阶段的可信原则下的可信证据主要为设计文档化，即过程设计文档化、结构设计文档化。

4.4.5.4 编码阶段

编码阶段的可信原则下的可信证据主要为源代码文档化，即程序内部文档化、数据说明文档化。

4.4.5.5 测试阶段

测试阶段的可信原则下的可信证据主要为测试文档化，即测试计划书、集成测试文档化、模块测试文档化、调试文档化、验收测试文档化。

4.4.6 可验证过程域的可信证据

4.4.6.1 概述

可验证过程域共有 8 个可信原则，涉及环境、需求、设计、编码、和测试五个阶段，共包含 24 个可信证据。

4.4.6.2 环境阶段

环境阶段的可信要求下的可信证据主要为形式化验证要求，即形式化可跟踪性、形式化验证同行评审。

4.4.6.3 需求阶段

需求阶段可信原则下的可信证据如下：

a) 形式化需求验证：形式化需求范围、需求形式化程度、形式化需求规约；

b) 需求分析评审：操作概念和场景的评审、软件需求的正式评审与审查、需求获取阶段的用户介入、需求审计确认。

4.4.6.4 设计阶段

设计阶段可信原则下的可信证据如下：

a) 形式化设计验证：形式化设计验证范围、形式化设计验证、和形式化设计规约；

b) 设计评审：设计阶段评审组的经验、设计阶段的同行评审、设计审计策略。

4.4.6.5 编码阶段

编码阶段可信下的可信证据如下：

a) 形式化代码验证：形式化代码验证范围、形式化代码验证；

b) 源代码评审：源码评审标准、代码评审的程度、代码评审人员的经验。

4.4.6.6 测试阶段

测试阶段的可信原则下的可信证据主要为测试评审，即测试环境的评审、测试计划的评审、测试用例的评审、测试标准的遵从。

4.4.7 可追溯过程域的可信证据

4.4.7.1 概述

可追溯域共有 4 个可信原则，涉及需求、设计、编码和测试四个开发阶段，共包含 10 个可信证据。

4.4.7.2 需求阶段

需求阶段的可信原则下的可信证据主要为需求可跟踪性：软件制品到需求的可跟踪性等级。

4.4.7.3 设计阶段

设计阶段的可信原则下的可信证据主要为设计可跟踪性：设计到需求的追踪程度、需求到设计的追踪程度。

4.4.7.4 编码阶段

编码阶段的可信原则下的可信证据主要为源代码可跟踪性，即需求到原始需求和利益相关方的追踪程度、需求到源代码的可追踪程度、源代码到设计的可追踪程度、设计到源代码的可追踪程度。

4.4.7.5 测试阶段

测试阶段的可信原则下的可信证据主要为测试可跟踪性：测试到需求的可追踪性、测试到设计的可追踪性、测试到源代码的可追踪性。

4.5 软件制品可信证据

4.5.1 概述

软件制品可信证据共 84 个，隶属于 8 个质量特性，覆盖软件开发全生存周期的制品。每个质量特

性由一组相关子特性组成，而每个子特性包含若干个证据。制品可信原则按照软件开发阶段和质量特性所对应的两种等级分布，分别见表6和表7。

表6　按照软件开发阶段表示制品证据的等级分布

开发阶段	证据场景等级					
	一级	二级	三级	四级	五级	总计
设计	1	3	4	6	0	14
编码	1	0	1	0	0	2
测试	0	5	12	5	1	23
交付	0	3	17	11	1	32
维护	0	0	4	0	0	4
软件开发全过程	8	1	0	0	0	9
证据数总计	10	12	38	22	2	84

表7　按照质量特性分类表示制品证据的等级分布

质量特性	相关子特性	证据场景等级					
		一级	二级	三级	四级	五级	证据数总计
功能性	4	3	0	8	1	1	13
性能效率	4	1	0	3	8	0	12
兼容性	3	1	4	0	0	0	5
易用性	6	1	3	7	1	0	12
可靠性	5	1	2	6	3	0	12
信息安全性	6	1	0	9	2	0	12
维护性	6	1	1	4	6	0	12
可移植性	3	1	2	1	1	1	6
证据数总计	37	10	12	38	22	2	84

表6和表7中的证据场景指该证据在所属的场景等级需要。在证据场景等级的界定方面借鉴了产品失效严重性分级体系，根据软件产品可信程度下降可能导致的风险大小，从最低风险等级的一级（原型与实验级）到最高风险等级五级（生命攸关级）逐级划分见表8。

表8　产品失效严重性分级表

级别类型	一级	二级	三级	四级	五级
系统级别	原型与实验级	实用工具级	一般产品级	规模商业级	生命攸关级
失效损失级别	微小损失	可接受、可恢复的损失	造成较大损失	大规模危害性严重损失	灾难性损失
典型系统	实验室级别系统	办公室级别办公自动化软件	商业办公软件	金融系统（支付宝、网银系统）	航空航天高铁

具体证据见附录 C。

以下将从 8 个质量特性的角度，提出软件制品可信原则的证据要求。

4.5.2 功能性

4.5.2.1 概述

功能性质量特性共有 13 个证据，涉及设计、编码、测试、交付和软件开发全过程 5 个阶段。

4.5.2.2 设计阶段

设计阶段的质量特性下可信证据如下：

a) 功能完备性：设计验证；

b) 功能正确性：设计的正确与完整性。

4.5.2.3 编码阶段

编码阶段的质量特性下可信证据如下：

a) 功能依从性：代码中的注释率；

b) 功能正确性：错误检测。

4.5.2.4 测试阶段

测试阶段的质量特性下可信证据如下：

a) 功能完备性：需求覆盖率和功能实现覆盖率；

b) 功能正确性：功能正确性、计算正确性、估算的缺陷密度、实际的缺陷密度。

4.5.2.5 交付阶段

交付阶段的质量特性下可信证据如下：

a) 功能稳定性：功能稳定性；

b) 功能正确性：接口一致性。

4.5.2.6 软件开发全过程

软件开发全过程的质量特性下可信证据主要为功能性的依从性，包含一条证据：功能性的依从性。

4.5.3 性能效率

4.5.3.1 概述

性能效率质量特性共有 12 个证据，处于交付和软件开发全过程 2 个阶段。

4.5.3.2 交付阶段

交付阶段的质量特性下可信证据如下：

a) 时间特性：平均响应时间、响应时间的充分性、平均周转时间、周转时间的充分性、平均吞吐率；

b) 资源利用性：平均处理器利用性、平均内存利用性、平均 I/O 设备利用性、带宽利用性；

c) 容量：事务处理能力、并发访问量。

4.5.3.3 软件开发全过程

软件开发全过程的质量特性下可信证据主要为性能效率的依从性，包含一条证据：性能效率的依从性。

4.5.4 兼容性

4.5.4.1 概述

兼容性质量特性共有 5 个证据，涉及设计、测试和软件开发全过程 3 个阶段。

4.5.4.2 设计阶段

设计阶段的质量特性下可信证据如下：

a) 共存性：程序部件应用范围的广泛性；

b) 互操作性：通信通用性。

4.5.4.3 测试阶段

测试阶段的质量特性下可信证据如下：

a) 共存性：环境或资源共享性；

b) 互操作性：数据互连性。

4.5.4.4 软件开发全过程

软件开发全过程的质量特性下可信证据主要为兼容性的依从性，包含一条证据：兼容性的依从性。

4.5.5 易用性

4.5.5.1 概述

易用性质量特性共有 12 个证据，涉及设计、测试、交付、软件开发全过程 4 个阶段。

4.5.5.2 设计阶段

设计阶段的质量特性下可信证据主要为可操作性，包含一条证据：撤销能力。

4.5.5.3 测试阶段

测试阶段的质量特性下可信证据如下：

a) 可操作性：信息明确性、监控能力；

b) 用户错误防范机制：抵御误操作。

4.5.5.4 交付阶段

交付阶段的质量特性下可信证据如下：

a) 易用性的描述：描述的完整性和正确性；

b) 易识别性：描述完整性、示范覆盖率、站点自描述能力；

c) 易学性：用户指导完整性、错误理解率；

d) 用户错误防范机制：用户错误的修复。

4.5.5.5 软件开发全过程

软件开发全过程的质量特性下可信证据主要为易用性的依从性，包含一条证据：易用性的依从性。

4.5.6 可靠性

4.5.6.1 概述

可靠性质量特性共有 12 个证据，涉及设计、测试、交付、维护和软件开发全过程 5 个阶段。

4.5.6.2 设计阶段

设计阶段的质量特性下可信证据如下：

a) 成熟性：是否进行可靠性关键组件识别、是否进行关键组件可靠性分析；

b) 容错性：关键组件冗余度。

4.5.6.3 测试阶段

测试阶段的质量特性下可信证据主要为成熟性，包含一条证据：测试覆盖度。

4.5.6.4 交付阶段

交付阶段的质量特性下可信证据主要为易恢复性，包含两条证据：平均恢复时间、数据备份的完成度。

4.5.6.5 维护阶段

维护阶段的质量特性下可信证据如下：

a) 成熟性：故障频率、平均故障间隔时间；

b) 可用性：平均宕机时间、系统可用性。

4.5.6.6 软件开发全过程

软件开发全过程的质量特性下可信证据如下：

a) 成熟性：缺陷修复；

b) 可靠性的依从性：可靠性的依从性。

4.5.7 信息安全性

4.5.7.1 概述

信息安全性质量特性共有 12 个证据，涉及设计、测试、交付和软件开发全过程 4 个阶段。

4.5.7.2 设计阶段

设计阶段的质量特性下可信证据如下：

a) 保密性：数据加密正确性、密码算法的强度；

b) 抗抵赖性：唯一/特定身份鉴别方式的使用；

c) 可核查性：用户审计跟踪的完整性、系统日志的保留。

4.5.7.3 测试阶段

测试阶段的质量特性下可信证据如下：

a) 保密性：访问可控性；

b) 真实性：验证规则的一致性。

4.5.7.4 交付阶段

交付阶段的质量特性下可信证据如下：

a) 完整性：数据被破坏性、数据破坏保护、数据完整性保护；

b) 真实性：验证机制的充分性。

4.5.7.5 软件开发全过程

软件开发全过程的质量特性下可信证据主要为信息安全性的依从性，包含一条证据：信息安全性的依从性。

4.5.8 维护性

4.5.8.1 概述

维护性质量特性共有 12 个证据，涉及设计、测试、交付和软件开发全过程 4 个阶段。

4.5.8.2 设计阶段

设计阶段的质量特性下可信证据主要为模块化，包含一条证据：组件耦合度。

4.5.8.3 测试阶段

测试阶段的质量特性下可信证据如下：

a) 模块化：充分的圈复杂度；

b) 可重用性：资产可重用性、编码规则一致性；

c) 易分析性：诊断功能的有效性、诊断功能的充分性；

d) 易修改性：修改的效率；

e) 易测试性：测试功能的完整性、测试的自主性、测试的可重启性。

4.5.8.4 交付阶段

交付阶段质量特性下可信证据主要为易修改性，包含一条证据：修改的正确性。

4.5.8.5 软件开发全过程

软件开发全过程的质量特性下可信证据主要为维护性的依从性，包含一条证据：维护性的依从性。

4.5.9 可移植性

4.5.9.1 概述

可移植性质量特性共有 6 个证据，处于交付和软件开发全过程 2 个阶段。

4.5.9.2 交付阶段

交付阶段的质量特性下可信证据如下：

a) 适应性：系统软件环境的适应性、硬件环境的适应性、操作环境的适应性；

b) 易安装性：易安装性（安装的灵活性）、安装时间的效率。

4.5.9.3 软件开发全过程

软件开发全过程的质量特性下可信证据主要为可移植性的依从性，包含一条证据：可移植性的依

从性。

5 软件可信度等级

5.1 软件过程可信度等级

5.1.1 概述

软件过程可信度等级由一组可信原则和原则对应的证据组成，共有 5 个可信度等级，见图 2。

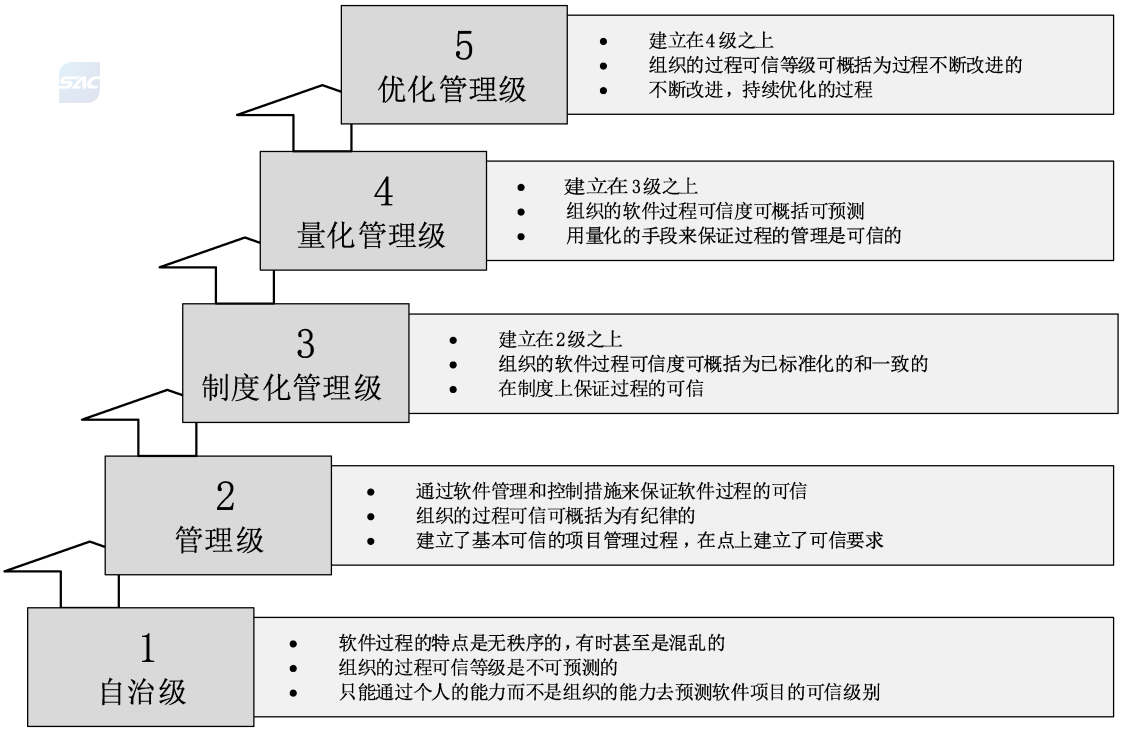

图 2 过程可信度等级

5.1.2 过程可信度等级 1：自治级

自治级表现特征包括但不限于：

a) 软件过程的特点是无秩序的，甚至是混乱的；
b) 软件开发组织不能够提供开发和维护软件的稳定环境；
c) 在开发中遇到危机时，一般项目组就会抛弃临时拼凑的软件开发过程，恢复到仅作代码测试；
d) 项目的成功完全依赖于个人或团队；
e) 进度、预算、功能性和产品质量不可预测；
f) 实施情况依赖于个人的能力，随着个人技能、知识和动机的不同而变化，只能通过个人的能力而不是组织的能力去预测软件项目的可信级别。

5.1.3 过程可信度等级 2：管理级

管理级表现特征包括但不限于：

a) 软件开发组织已建立了基本可信的项目管理过程，通过软件管理和控制措施来保证软件过程的可信；
b) 在点上建立了可信要求，可用于对成本、进度和功能特性进行可信度跟踪；

c) 软件项目管理过程可信制度化，组织能重复在以前类似项目中的成功实践；
d) 对项目实际可行的软件管理和控制措施应根据以前项目总结出的经验和当前项目的实际需求而制定，进而保证其可信性。

5.1.4 过程可信度等级3：制度化管理级

制度化管理级表现特征包括但不限于：
a) 用于管理的和工程的软件过程均已文档化、标准化，并形成了整个软件组织的可信的标准过程；
b) 组织的全部项目均采用与实际情况相吻合的、适当修改后的标准软件过程来进行操作；
c) 软件组织通过已标准化的过程来保证对过程管理的控制和跟踪是可信的；
d) 项目的过程能力是建立在整个组织的标准软件过程的基础上的，组织范围内对项目定义的软件过程中的活动、角色和职责具有共同的和一致的理解，进而保证了过程的可信。

5.1.5 过程可信度等级4：量化管理级

量化管理级表现特征包括但不限于：
a) 软件开发组织对软件产品和过程设置了定量的质量目标，并经常对此进行测量和检查；
b) 软件过程均已配备有妥善定义的和一致的测量。

5.1.6 过程可信度等级5：优化管理级

优化管理级表现特征包括但不限于：
a) 软件过程在可管理、可验证和可追溯上进一步加强，是一个持续优化的过程；
b) 软件组织为提高其软件过程可信度等级进行着不懈的努力。

5.2 软件制品可信度等级

5.2.1 概述

软件制品可信度等级与软件过程可信度等级是有联系的，制品可信度等级的达成除与对应的制品证据可信度等级有关，也对软件过程可信度等级有要求，具体见表9。

表9 制品可信度等级要求

制品可信度等级	过程可信度等级要求	制品证据可信度等级要求
制品可信度等级1	1级	1级
制品可信度等级2	1级以上	1级，2级
制品可信度等级3	1级以上	1级，2级，3级
制品可信度等级4	2级以上	1级，2级，3级，4级
制品可信度等级5	2级以上	1级，2级，3级，4级，5级

5.2.2 制品可信度等级1

制品可信度等级1表现特征包括但不限于：
a) 在可信度等级1的基础上，软件组织的过程仅需要满足一些制品的证据要求。这些制品是软件开发最基本的；
b) 过程可信度等级要求达到1级。

5.2.3 制品可信度等级 2

制品可信度等级 2 表现特征包括但不限于：

a) 软件开发过程中的制品需同时满足 1 级和 2 级证据的可信要求；
b) 过程可信度等级要求达到 1 级以上；
c) 制品证据在兼容性、易用性、可靠性、可维护性和可移植性方面有增强。

5.2.4 制品可信度等级 3

制品可信度等级 3 表现特征包括但不限于：

a) 软件组织已经达到了第 1、2 和 3 级要求的制品证据的可信要求；
b) 过程可信度等级要求达到 1 级以上；
c) 制品证据在功能性、性能效率、易用性、可靠性、信息安全性、可维护性和可移植性上有可信的进一步要求。

5.2.5 制品可信度等级 4

制品可信度等级 4 表现特征包括但不限于：

a) 软件组织达到了第 1、2、3 和 4 级的各个制品证据规定的可信要求；
b) 过程可信度等级要求达到 2 级以上；
c) 制品证据在功能性、性能效率、易用性、可靠性、信息安全性、可维护性和可移植性上定义了更高的可信要求。

5.2.6 制品可信度等级 5

制品可信度等级 5 表现特征包括但不限于：

a) 软件组织达到了第 1、2、3、4 和 5 级的各个制品证据规定的可信要求；
b) 过程可信度等级要求达到 2 级以上；
c) 制品证据在功能性和可移植性上也有进一步要求。

6 软件过程可信度评估

6.1 软件过程可信度评估过程

基于软件开发过程中数据所形成的证据，量化评估软件过程可信程度的模型，可有效支持软件过程可信度评估的客观性和全面性。软件组织或项目可按以下 6 个步骤对软件过程可信度进行评估：

a) 确定软件开发要求的过程可信级别；
b) 基于模型的证据要求和项目实际情况，对证据进行裁剪，建立组织或项目适用的证据集合；
c) 收集证据；
d) 评估证据达到的可信等级。对于未达到期望等级的证据所关联的活动，应采取适当的纠正措施，以管理和控制后续过程活动，保证整体目标的达成；
e) 评估可信原则达到的可信等级；
f) 评估软件过程的实际可信级别。

软件过程可信度评估示例参见附录 D。

6.2 过程证据的可信度评估

过程证据可信度的评估方法如下所述。

设 T-Level 表示某证据的可信等级，则：

——T-Level＝1：若该证据的测量值小于该证据的最小可信等级，则该证据的可信等级为 1；

——T-Level＝X(1＜X≤5)：若该证据的测量值介于其预定的最小可信等级和最大可信等级之间，则 X 表示其可映射到的可信等级；

——T-Level＝T-Max：若该证据的测量值大于或等于该证据的最大可信等级，其可信等级即为最大可信等级。

6.3 可信原则的可信度评估

在本模型中，可信原则满足的可信程度取决于其下属的证据所表征的可信级别。

可信原则达到的可信级别的评估方法如下所述。

初始状态下，可信原则的可信度为 1。

设 S-Level 表示某可信原则的可信等级，则：

——S-Level＝2：该原则下所有的 2 级证据，这些证据的可信等级都大于或等于 2；

——S-Level ＝3：该原则下所有的 2 级和 3 级证据，这些证据的可信等级：T-Level≥3 或者 T-Level＝2 并且 2 级是该证据的最大可信等级；

——S-Level ＝4：该原则下所有的 2 级、3 级和 4 级证据，这些证据的可信等级：T-Level≥4 或者 T-Level＝T-Max 达到该证据的最大可信等级；

——S-Level ＝5：该原则下所有的 2 级、3 级、4 级和 5 级证据，这些证据的可信等级：T-Level＝5 或者 T-Level＝T-Max 达到该证据的最大可信等级。

6.4 软件过程的可信度评估

基于对可信原则满足程度的评估，进一步评估整个软件过程实现的可信度水平。指定的可信范围内，所有适用的原则应达到要求的可信级别。

7 软件制品可信度评估

7.1 软件制品可信度评估过程

基于软件开发过程中数据所形成的证据，评估软件制品的可信程度。软件组织或项目可按以下 5 个步骤对软件制品可信度进行评估：

a) 确定软件开发要求的制品可信级别。

b) 基于模型的证据要求和项目实际情况，进行证据裁剪，建立组织或项目适用的证据集合。

c) 收集证据。

d) 评估证据是否达到要求。对于未达到期望等级的证据所关联的制品，应采取适当的纠正措施，保证整体目标的达成。

e) 结合过程可信度评估等级，评估软件制品的实际可信度等级。

软件制品可信度评估示例参见附录 D。

7.2 制品证据的可信度评估

制品证据可信度的评估方法如下所述。

设 T-level 表示某制品证据的可信等级，则：

——T-level＝1：若该证据的测量值在该证据需要满足的阈值之外，则该证据没达到其适用的可信等级 X，其可信度等级为 1；

——T-level＝X：若该证据的测量值在需要满足的阈值范围内，则该证据达到其适用的可信等级

X，其可信度等级为 X。

7.3 软件制品的可信度评估

基于对制品证据满足程度的评估，进一步评估整个软件制品的可信度水平。对软件制品的可信度评估遵循木桶原理，即在指定的可信范围内，所有适用的原则应达到要求的可信级别。

初始状态下，软件制品的可信度为 1。

设 A-Level 表示软件制品的可信等级，则：

——A-Level＝2：所有的 2 级制品证据都达到了 2 级要求，即 A-level＝2；

——A-Level＝3：在 2 级的基础上，所有的 3 级制品证据都达到了 3 级要求，即 A-level＝3；

——A-Level＝4：在 3 级的基础上，所有的 4 级制品证据都达到了 4 级要求，即 A-level＝4；

——A-Level＝5：在 4 级的基础上，所有的 5 级制品证据都达到了 5 级要求，即 A-level＝5。

对于在模型应用时判定为裁剪的制品证据，不做评估。

附 录 A
（资料性附录）
可信原则与 CMMI 过程域

CMMI 模型共有 22 个过程域，分为 5 个成熟度等级，表 A.1 详细标注了每个过程域的名称和成熟度等级。CMMI 强调过程管理的制度化和过程的稳定性，通过过程的改进不断使得过程稳定，并且在控制下持续改善能力。随着成熟能力的提高，过程量化控制的能力越强。软件可信的目标并不在于强调过程的稳定可控，而是过程管理的可信性。稳定的过程在测量分析的能力上具备更好的可信程度，但并不意味在所有的可信要求上都具备更好的可信程度。

表 A.1 CMMI 模型的 22 个过程域

等级	过程域(Process Area)
5. 优化级	OPM(Organizational Performance Management：组织绩效管理) CAR(Causal Analysis and Resolution：原因分析与解决)
4. 量化级	OPP[Organizational Process Performance：组织过程性能(绩效)] QPM(Quantitative Project Management：量化项目管理)
3. 定义级	RD(Requirement Development：需求开发) TS(Technical Solution：技术解决方案) PI(Product Integration：产品集成) VER(Verification：验证) VAL(Validation：确认) OPF[Organizational Process Focus：组织过程关注(焦点)] OPD(Organizational Process Definition：组织过程定义) OT(Organizational Training：组织培训) IPM(Integrated Project Management：集成项目管理) RSKM(Risk Management：风险管理) DAR(Decision Analysis and Resolution：决策分析与解决)
2. 管理级	REQM(Requirement Management：需求管理) PP(Project Planning：项目计划) PMC(Project Monitoring and Control：项目监督与控制) SAM(Supplier Agreement Management：供应商协议管理) MA(Measurement and Analysis：测量和分析) PPQA(Process and Product Quality Assurance：过程和产品质量保证) CM(Configuration Management：配置管理)
1. 初始级	—

为了叙述简便，所有 22 个过程域 PA 均采用英文首字母简写的方式表示。

可信原则与 CMMI 过程域的对应关系见图 A.1，其中实心圆点表示可信原则与过程域 PA 存在主要对应关系，而空心圆点表示该可信原则应该涵盖的过程域 PA，但实际并未与其存在主要对应关系。

可信原则覆盖 CMMI V1.3 包含的 22 个过程域，并扩充了对可信实体、开源软件的支持，以及对过程制品的可信增强。增加了实体安全类中的 4 个可信原则，扩充了 CMMI 模型在实体资源的实践要求。可信实体的保障则是由“实体安全可信”域的四个可信原则来保障。CMMI 模型虽然也包括软件

开发实体(实践者)的活动以及工作环境的建立,譬如 OPD 中对过程和环境的要求,以及 OT 要求的培训,但关注的主要焦点是工程师的技能是否可以胜任工作以及人员培训、沟通、协同和承诺等,不能完全覆盖本模型对实体可信的全部要求。此外,开源软件是当前软件开发组织中最常用到的资源,在 CMMI 模型中没有专门的关注。

No	可信原则 \ CMMI PA	OPD	OPF	OPM	OPP	OT	IPM	PMC	PP	QPM	REQM	RSKM	SAM	PI	RD	TS	VAL	VER	CAR	CM	DAR	MA	PPQA
1	安全政策	●				●																	
2	安全管理策略	●				●																	
3	安全环境	●																		●			
4	信息安全工具支持	●																					
5	开发环境	●																					
6	重用支持													●	●	●							
7	开源支持																						
8	采购支持												●										
9	需求分析工具														●								
10	设计工具															●		●					
11	源代码分析工具																	●					
12	测试工具																●	●					
13	知识共享	●	●			●																	
14	管理制度化	●		○		●	○		○	○											○		
15	环境完整性	●																					
16	风险管理											●											
17	计划						●		●														
18	度量分析	●			●					●									●			●	
19	配置管理																			●			
20	过程审计																						●
21	需求分析文档化													●	●					●			
22	设计文档化													●		●				●			
23	源代码文档化													●		●				●			
24	测试文档化													●	●					●			
25	形式化验证要求																●						
26	形式化需求验证																●						
27	需求分析评审																	●					
28	设计评审																	●					
29	形式化设计验证																●						
30	形式化代码验证																●						
31	源代码评审																	●					
32	测试评审																	●					
33	需求可跟踪性							●			●												
34	设计可跟踪性							●			●												
35	源代码可跟踪性							●			●												
36	测试可跟踪性							●			●												

图 A.1　可信原则与 CMMI 过程域的对应关系

附 录 B
（规范性附录）
软件过程证据

B.1 实体安全过程域的过程证据

实体安全过程域共有 14 个过程证据，全部贯穿软件开发全过程，见表 B.1。

表 B.1 实体安全过程域的过程证据

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
软件开发全过程	安全政策	安全政策的建立	是否建立安全政策	布尔型	2	是	同二级要求	同二级要求	同二级要求
		安全相关政策的了解	人员是否了解安全相关政策	布尔型	2	是	同二级要求	同二级要求	同二级要求
		培训的开展	是否对组织成员进行过软件可信包括安全方面的培训	布尔型	3	—	是	同三级要求	同三级要求
软件开发全过程	安全管理策略	背景调查	人员入职前是否做过背景调查	布尔型	2	是	同二级要求	同二级要求	同二级要求
		人员对可信知识了解程度	组织成员对于软件可信知识的了解程度	等级型	2	2 一般了解	3 有专门的培训	4 有专业的职业认证资格	同四级要求
		信息安全管理的程度	组织中针对信息安全管理的程序	等级型	2	2 建立了规程	3 建立专门的信息安全体系	4 通过信息安全体系审核	同四级要求
		项目遵循安全要求的程度	组织中，项目遵循安全要求的程度	等级型	2	2 遵照执行，但少数有弱项	3 遵照执行，但多数有弱项	4 遵照执行，没有弱项	同四级要求
		安全管理相关的培训的开展	是否开展了安全管理相关的培训	布尔型	3	—	是	同级三要求	同三级要求
		共享监控机制	对于知识共享的方式、质量和知识流动是否有监控机制	布尔型	3	—	是	同三级要求	同三级要求

表 B.1（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
软件开发全过程	安全环境	安全工作环境标准的建立	是否建立安全工作环境标准	布尔型	2	是	同二级要求	同二级要求	同二级要求
		软件过程和资产访问权限和环境的建立	是否建立软件过程和资产访问权限和环境	布尔型	2	是	同二级要求	同二级要求	同二级要求
		网络服务安全性	常见的网络服务有无可信信道来提供加密、认证或完整性等保护	等级型	2	2 少部分保护	3 大部分保护	4 全部保护	同四级要求
软件开发全过程	信息安全工具支持	自动化工具对权限控制的支持程度	按照权限控制工具的自动化程度分级	等级型	2	2 少部分支持	3 大部分支持	4 全部支持	同四级要求
		组织结构对权限支持	组织结构是否对权限支持	布尔型	3	—	是	同三级要求	同三级要求
注：“—”表明该证据在该等级不做要求。									

B.2 开发支持过程域的过程证据

开发支持过程域共有 33 个过程证据，涉及环境、需求、设计、编码、和测试五个阶段，见表 B.2。

表 B.2 开发支持过程域的过程证据

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
环境	开发环境工具	开发工具支持	是否有开发工具支持	布尔型	2	是	同二级要求	同二级要求	同二级要求
		工作环境标准的建立	是否建立了工作环境标准	布尔型	3	—	是	同三级要求	同三级要求
		环境和工具培训的开展	是否开展了环境和工具的培训	布尔型	3	—	是	同三级要求	同三级要求

表 B.2（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
环境	重用支持	重用准则的建立	是否建立了重用的准则	布尔型	2	是	同二级要求	同二级要求	同二级要求
		重用分析	分析构建或使用重用模块的成本和风险	等级型	2	2 低成本风险	3 一般	4 高成本风险	同四级要求
		支持和维护重用的软件架构的开发	开发支持和维护重用的软件架构的程度	等级型	2	2 建立了重用部件支持文档	3 建立支持重用的软件体系架构	4 建立自动化工具支持	同四级要求
		重用部件的质量评价	对重用部件质量评价的程度	等级型	2	2 通过了系统测试	3 交付使用	同三级要求	同三级要求
		可重用部件设计标准的建立	是否建立了可重用部件的设计标准	布尔型	3	—	是	同三级要求	同三级要求
环境	开源支持	对选择的开源软件的调研	是否对选择开源软件进行过调研	布尔型	2	是	同二级要求	同二级要求	同二级要求
		对采用开源软件的成本估算	是否对采用开源软件的成本估算	布尔型	2	是	同二级要求	同二级要求	同二级要求
		开源软件提供方持续的技术支持	对开源软件提供方持续的技术支持程度	等级型	2	2 没有	3 一般	4 强	同四级要求
		开源软件的质量评价	对开源软件的质量评价的要求	等级型	2	2 内部评价	3 开源社区评价	4 可信评估	同四级要求
		采用开源软件的风险评估	是否采用开源软件的风险评估	布尔型	3	—	是	同三级要求	同三级要求
环境	采购支持	供应商目录的建立	是否建立了供应商目录	布尔型	2	是	同二级要求	同二级要求	同二级要求
		采购合同的建立与维护	是否建立和维护了采购合同	布尔型	2	是	同二级要求	同二级要求	同二级要求
		采购产品质量的评价	评价采购产品质量等级	等级型	2	2 验收测试采购产品	3 对采购产品进行了可信评估	同三级要求	同三级要求

表 B.2（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
需求	需求分析工具	人员对需求分析工具的掌握程度	根据需求分析工具在组织内的使用范围、使用时间以及开发人员掌握程度分级	等级型	2	2 开发小组，半年以上，一般掌握	3 部门内，使用超过1年，较熟练	同三级要求	同三级要求
		软件需求分析自动化环境	对软件需求分析自动化程度的要求	等级型	2	2 需求分析工具覆盖少量需求开发和管理活动	3 需求分析工具覆盖部分需求开发管理活动	4 需求分析工具覆盖关键的需求规约、一致性检查、需求跟踪和文档生成活动	5 需求分析工具全面覆盖需求规约、一致性检查、需求跟踪和文档生成
		自动化工具对需求分析的支持程度	按照需求分析工具的自动化程度分级	等级型	2	2 少部分支持	3 大部分支持	4 全部支持	同四级要求
设计	设计工具	人员对设计工具的掌握程度	根据设计工具在组织内的使用范围、使用时间以及开发人员掌握程度分级	等级型	2	3 开发小组，半年以上，一般掌握	同二级要求	4 部门内，使用超过1年，较熟练	同四级要求
		设计自动化环境	使用集成文本、标准软件图形和数据字典的计算机化的工具	等级型	2	2 只有文本自动化工具支持的半正式设计	3 具有文本/图形工具支持的半正式设计	4 正式的设计方法和自动化的文本/图形支持	同四级要求
编码	源代码分析工具	人员对代码分析工具的掌握程度	根据代码分析工具在组织内的使用范围、使用时间以及开发人员掌握程度分级	等级型	2	2 个人使用，半年以内，不太熟练	3 开发小组，半年以上，一般掌握	4 部门内，使用超过1年，较熟练	同四级要求
		自动化工具对源代码分析的支持程度	按照源代码分析工具的自动化程度分级	等级型	2	2 有工具	3 工具可部署自动代码分析	同三级要求	同三级要求
		源代码库	对于正在开发的源代码组件进行系统化的存储和控制变更的手段	等级型	2	2 有源码库	3 源码库有合理的配置管理策略	4 源码库可支持代码分析工具	同四级要求

表 B.2（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
测试	测试工具	测试工具类别的完整性	根据组织使用的测试工具的类型（实施软件标准类、需求验证类、产品管理类、错误检测和性能分析类、产生测试脚本类），将其完整级别分级	等级型	2	2 完整性中（包括类别数≥3）	同二级要求	3 完整性高（包括类别数≥4）	同四级要求
		测试环境的搭建	根据测试环境的完整性分级	等级型	2	3 有少量遗漏	4 较为完整	5 完整	同四级要求
		人员对测试工具的掌握程度	根据测试工具在组织内的使用范围、使用时间以及开发人员掌握程度分级	等级型	2	3 开发小组，半年以上，一般掌握	4 部门内，使用超过1年，较熟练	同三级要求	5 全公司范围、使用超过3年，熟练掌握
		测试数据和测试用例的设计	根据测试数据和测试用例的覆盖程度分级	等级型	2	3 有少量遗漏	4 覆盖程度较高	5 覆盖程度高	同四级要求
		自动化工具对软件测试的支持程度	按照测试工具的自动化程度分级	等级型	2	2 少部分支持	同二级要求	同二级要求	3 大部分支持
		单元测试策略	是否有单元测试策略	布尔型	3	—	是	同三级要求	同三级要求
		集成测试策略	是否有集成测试策略	布尔型	3	—	是	同三级要求	同三级要求
		配置项测试策略	是否有配置项测试策略	布尔型	3	—	是	同三级要求	同三级要求
		组件测试策略	是否有组件测试策略	布尔型	3	—	是	同三级要求	同三级要求
注：“—”表明该证据在该等级不做要求。									

B.3 可管理过程域的过程证据

可管理过程域共有 41 个过程证据，涉及环境、软件开发全过程，见表 B.3。

表 B.3 可管理过程域的过程证据

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
环境	知识共享	支持知识共享的组织资产库覆盖的范围	支持知识共享的组织资产库覆盖的范围的要求	等级型	2	2 有知识共享	3 在组织级别建立了支持知识共享的资产库	同三级要求	同三级要求
		共享知识的获取方式	组织对获取共享知识的方式的要求	等级型	2	2 随机获取	3 建立常态的收集改进建议和共享知识的机制	4 对重要建议有计划、实验、评价和部署的过程	5 对部署的改进有计划、实验、评价和部署的过程
		组织资产的发改进与维护	改进和维护组织资产的程度	等级型	3	—	3 计划、推广和监督改进的过程资产	4 基于过程资产。量化分析持续的改进	同四级要求
		组织培训能力的建立和维护	是否具备建立和维护组织的培训能力	布尔型	3	—	是	同三级要求	同三级要求
环境	管理制度化	管理规程的建立	是否建立管理规程	布尔型	2	是	同二级要求	同二级要求	同二级要求
		职责和权限的明确	能否明确职责和权限	布尔型	2	是	同二级要求	同二级要求	同二级要求
		培训的开展	是否开展了培训	布尔型	2	是	同二级要求	同二级要求	同二级要求
		管理规程的覆盖范围	组织中对管理规程的覆盖范围的要求	等级型	2	2 建立项目估算、计划的基本程序	3 建立了 CMMI 3 级所有 PA 的管理规程	4 建立了量化管理过程的规程	5 建立改进组织业务性能的管理规程
		决策过程管理规程的建立	是否建立决策过程的管理规程	布尔型	3	—	是	同三级要求	同三级要求

表 B.3（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
环境	环境完整性	软件工程环境变更权限的配置	是否定义软件工程环境变更权限	布尔型	2	是	同二级要求	同二级要求	同二级要求
		软件工程环境变更的记录	是否有软件工程环境组件变更识别的功能	布尔型	3	—	是	同三级要求	同三级要求
		软件工程环境的变更审查	是否对软件工程环境变更的审查	布尔型	3	—	是	同三级要求	同三级要求
软件开发全过程	风险管理	风险管理计划	是否有风险管理计划	布尔型	2	是	同二级要求	同二级要求	同二级要求
		发现识别和评价标准和指南的建立	是否建立了发现识别和评价的标准和指南	布尔型	3	—	是	同三级要求	同三级要求
		风险管理策略的建立	是否建立风险管理的策略	布尔型	3	—	是	同三级要求	同三级要求
		项目风险管理的效果	项目风险管理的效果等级	等级型	3	—	3 识别的高等级风险少量发生	4 识别的高等级风险无发生	同四级要求
软件开发全过程	计划	软件开发计划书的编制	是否编制了软件开发计划书	布尔型	2	是	同二级要求	同二级要求	同二级要求
		项目的人员配置和利益相关方的明确	是否明确了项目的人员配置和利益相关方	布尔型	2	是	同二级要求	同二级要求	同二级要求
		项目过程制品和里程碑的明确定义与配置管理	项目过程制品和里程碑是否明确定义并建立合适的配置管理	布尔型	2	是	同二级要求	同二级要求	同二级要求
		风险的识别和评估	是否识别和评估了风险	布尔型	2	是	同二级要求	同二级要求	同二级要求
		项目计划的成熟程度	组织中，项目计划的成熟程度	等级型	2	2 项目做了估算	3 计划变更受控	4 计划执行无重大偏差	5 计划执行在统计意义上受控
		项目利用与贡献组织资产的程度	组织中，项目利用和贡献组织资产的程度	等级型	2	2 参考组织资产进行估算	3 基于组织过程建立项目过程	4 项目的产出和知识贡献到组织资产	同四级要求

表 B.3（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
软件开发全过程	测量与分析	测量目标的建立	是否建立了测量目标	布尔型	2	是	同二级要求	同二级要求	同二级要求
		相关的测量元、数据采集，以及测量和存储方法的建立	是否建立了相关的测量元、数据采集，以及测量和存储方法	布尔型	2	是	同二级要求	同二级要求	同二级要求
		测量体系覆盖的程度	组织中测量体系覆盖的程度	等级型	2	2 有测量	3 在组织级别建立了测量体系	4 建立了关键过程的性能基线和量化管理的方法	5 建立原因分析和改进效果评价的测量方法
		项目开展测量分析的程度	组织中，项目开展测量分析的程度	等级型	2	2 部分项目有测量	3 全部项目按组织要求进行了测量	同三级要求	同三级要求
		测量分析结果在组织级别的保存和分析	测量分析结果是否在组织级别进行保存和分析	布尔型	3	—	是	同三级要求	同三级要求
		测量数据的采集、分析和存储的工具支持	是否有工具支持测量数据的采集、分析和存储	布尔型	3	—	是	同三级要求	同三级要求
		选择的关键过程控制属性在统计意义上的情况	选择的关键过程控制属性是否在统计意义上稳定	布尔型	4	—	—	是	同四级要求
		过程异常地分析和改进	是否对过程异常进行了分析和改进	布尔型	4	—	—	是	同四级要求
		过程和产品持续改进机会的量化分析	是否通过量化方法分析过程和产品持续改进的机会	布尔型	5	—	—	—	是
	配置管理	配置管理策略	配置管理策略的建立情况	等级型	2	2 建立了配置管理策略	3 配置管理策略分区有序，利用工作和资产管理	同三级要求	同三级要求

表 B.3（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
软件开发全过程	配置管理	配置基线管理	配置基线的管理情况	等级型	2	2 识别了配置项，建立了配置基线	3 配置基线设置合理	同三级要求	同三级要求
		配置变更管理	是否进行了配置变更管理	布尔型	2	是	同二级要求	同二级要求	同二级要求
		配置状态审计	是否进行了配置状态审计	布尔型	2	是	同二级要求	同二级要求	同二级要求
		软件配置管理工具的使用	是否使用了软件配置管理工具	布尔型	2	是	同二级要求	同二级要求	同二级要求
		配置管理人员的成熟度	配置管理人员对配置管理工作的了解情况	等级型	2	2 了解配置职能	3 专职配置人员，熟练配置智能	同三级要求	同三级要求
软件开发全过程	过程审计	内控制度的有效性	控制系统资源的存取、控制系统资源的使用、建立按用户职能分配资源的制度、记录系统的使用情况、确认处理过程的准确性、管理人员对系统的修改、保护系统免遭计算机病毒的袭击	等级型	2	2 审计覆盖较少部分	3 审计覆盖较多部分	4 审计覆盖全部	同四级要求
		审计方法的充分性	包括面谈法、系统文档审阅法、观察法、计算机系统文字描述法、表格描述法、图形描述法	等级型	2	2 凭经验	3 有检查单等资源	4 有多种审计方式	同四级要求
		审计过程的规范性	包括准备阶段、实施阶段和报告阶段	等级型	2	2 有审计计划	3 有审计过程规程和指南	同三级要求	同三级要求
		审计数据	审计数据的完整程度	等级型	2	2 有审计记录	3 对审计进行测量分析	同三级要求	同三级要求
注：“—”表明该证据在该等级不做要求。									

B.4 文档化过程域的过程证据

文档化过程域共有 11 个过程证据，涉及需求、设计、编码和测试四个开发阶段，见表 B.4。

表 B.4 文档化过程域的过程证据

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
需求	需求分析文档化	系统需求文档化	系统需求文档应更详细的描述系统的功能性需求和非功能性需求。如果可以，非功能性需求的更进一步的信息可添加。根据系统需求内容的完整性分级	等级型	2	2 包含功能性需求和非功能性需求的描述	3 详细包含上面所描述的内容	同三级要求	同三级要求
		用户需求文档化	用户需求定义文档应描述系统为用户提供的服务，以及系统的非功能性需求。这些描述可使用自然语言、图表或者其他用户可以理解的形式。产品和过程标注也应被定义好。根据用户需求定义内容的完整性分级	等级型	2	2 至少要包含系统为用户提供服务的描述，对功能性需求的描述不是很完整和具体	3 详细包含上面所描述的内容	同三级要求	同三级要求
设计	设计文档化	过程设计文档化	根据是否完整、准确、具体地描述系统的过程设计，将过程设计文档化指标分级	等级型	2	2 刚好满足软件项目活动正常进行的需要	3 过程设计信息记录全面，能够充分满足软件项目活动需要	同三级要求	同三级要求
		结构设计文档化	根据是否完整、准确、具体地描述系统的体系结构设计，将体系结构设计文档化分级	等级型	2	2 刚好满足软件项目活动正常进行的需要	3 结构设计信息记录全面，能够充分满足软件项目活动的需要	同三级要求	同三级要求

表 B.4（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
编码	源代码文档化	程序内部文档化	程序内部文档包含恰当的标示符、适当的注释和程序的视觉组织等。根据这一概念，将程序内部文档化指标分级	等级型	2	2 程序内部文档化使得源程序代码恰好可读	3 程序内部的文档化使得源程序代码逻辑简明清晰、易读易懂	同三级要求	同三级要求
		数据说明文档化	数据说明的次序应标准化，使用了一个复杂的数据结构时宜用注解说明。根据这个原则，将数据说明文档化指标分级	等级型	2	2 刚好满足软件项目活动正常进行的需要	3 记录各种详细信息，能够充分满足软件项目活动的需要	同三级要求	同三级要求
测试	测试文档化	测试计划书	是否有测试计划书	布尔型	2	是	同二级要求	同二级要求	同二级要求
		集成测试文档化	根据是否对集成测试的方法选择、具体执行以及测试用例等信息进行详细说明，将集成测试文档化指标分级	等级型	2	2 刚好满足软件项目活动正常进行的需要	3 记录各种详细信息，能够充分满足软件项目活动的需要	同三级要求	同三级要求
		模块测试文档化	根据是否对模块接口、局部数据结构、重要执行通路、出错处理通路、边界条件的测试以及测试用例等信息进行详细说明，将模块测试文档化指标分级	等级型	2	2 刚好满足软件项目活动正常进行的需要	3 记录各种详细信息，能够充分满足软件项目活动的需要	同三级要求	同三级要求
		调试文档化	根据是否对调试的过程、调试的途径等信息进行详细说明，将验收测试文档化指标分级	等级型	2	2 刚好满足软件项目活动正常进行的需要	3 记录各种详细信息，能够充分满足软件项目活动的需要	同三级要求	同三级要求
		验收测试文档化	根据是否对验收测试中方法选择、具体执行情况、用户参与情况以及测试用例等信息进行详细说明，将验收测试文档化指标分级	等级型	2	2 包含总体功能描述或如何融入到原有系统中的其中一个	3 详细包含上面所描述的内容	同三级要求	同三级要求

B.5 可验证过程域的过程证据

可验证过程域共有 24 个过程证据，涉及环境、需求、设计、编码和测试五个阶段，见表 B.5。

表 B.5 可验证过程域的过程证据

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
环境	形式化验证要求	形式化可跟踪性	是否建立了从形式化需求规约——形式化设计——形式化设计验证——形式化代码验证的跟踪关系	布尔型	2	是	同二级要求	同二级要求	同二级要求
		形式化验证同行评审	在形式化规约和验证的活动中是否有同行评审	布尔型	2	是	同二级要求	同二级要求	同二级要求
需求	形式化需求验证	形式化需求范围	组织对形式化需求的要求范围	等级型	2	1 低(无形式化需求)	同二级要求	2 中(安全关键相关部分)	3 高(全系统)
		需求形式化程度	组织对需求形式化程度的要求	等级型	2	1 描述性需求	同二级要求	2 半形式化需求	3 形式化需求
		形式化需求规约	是否有形式化需求规约	布尔型	4	—	—	是	同四级要求
需求	需求分析评审	操作概念和场景的评审	是否周期性评审操作概念和场景	布尔型	2	是	同二级要求	同二级要求	同二级要求
		软件需求的正式评审与审查	指本项目组所进行的正式评审和审查的有效性。这些质量验证技术包括工作产品的小组评审，目的是判断其完整性，正确性，一致性和开发方针及标准的符合程度	等级型	2	3 评审和审查是非正式的，但是有效	4 绝大多数评审和审查都是正式的，非常有效的	5 所有评审和审查都是正式的，非常有效的	同四级要求
		需求获取阶段的用户介入	最终用户在需求收集阶段的介入程度	等级型	2	3 在需求获取阶段，用户在一定程度上介入	4 在需求获取阶段，大多数用户全面介入	5 所有用户都全面介入	同四级要求
		需求审计确认	是否有需求审计确认	布尔型	2	是	同二级要求	同二级要求	同二级要求

表 B.5（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
设计	设计评审	形式化设计验证范围	组织对形式化设计验证范围的要求	等级型	2	1 低(无形式化验证)	同二级要求	2 中(安全关键相关部分)	3 高(全系统)
		形式化设计验证	是否有形式化设计验证	布尔型	4	—	—	是	同四级要求
		形式化设计规约	是否有形式化设计规约	布尔型	4	—	—	是	同四级要求
设计	形式化设计验证	设计阶段评审组的经验	指项目组成员在执行正式设计评审方面的平均经验。这些从事质量验证技术，包括工作产品的小组评审，目的是判断完整性、正确性、一致性以及是否符合开发方针和标准	等级型	2	3 中等 平均 1 年	4 高 平均 2 年	同三级要求	同三级要求
		设计阶段的同行评审	是否有同行评审	布尔型	2	是	同二级要求	同二级要求	同二级要求
		设计审计策略	对设计阶段的过程产品是否有审计制度	布尔型	2	是	同二级要求	同二级要求	同二级要求
编码	形式化代码验证	形式化代码验证范围	形式化代码验证范围	等级型	2	1 低(无形式化验证)	同二级要求	2 中(安全关键相关部分)	3 高(全系统)
		形式化代码验证	是否有形式化代码验证	布尔型	5	—	—	—	是
编码	源代码评审	源码评审标准	是否有源码评审标准	布尔型	2	是	同二级要求	同二级要求	同二级要求
		代码评审的程度	组织对代码评审的程度的要求	等级型	2	2 同行评审	3 上级评审	4 管理评审	5 质量评审
		代码评审人员的经验	指项目组成员在执行代码审查方面的经验大小。这些从事质量验证技术，包括工作产品的小组评审，目的是判断完整性、正确性、一致性以及是否符合开发方针和标准	数值型	2	2 平均 1 年	3 平均 2 年	4 平均 3 年以上	同四级要求

表 B.5（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
测试	测试评审	测试环境的评审	是否对测试环境的评审	布尔型	2	是	同二级要求	同二级要求	同二级要求
		测试计划的评审	是否对测试计划的评审	布尔型	2	是	同二级要求	同二级要求	同二级要求
		测试用例的评审	是否对测试用例的评审	布尔型	2	是	同二级要求	同二级要求	同二级要求
		测试标准的遵从	测试满足的标准/计划中所有测试标准	百分比型	2	60%	80%	90%	95%
注：“—”表明该证据在该等级不做要求。									

B.6 可追溯过程域的过程证据

可追溯域共有 10 个过程证据，涉及需求、设计、编码和测试四个开发阶段，见表 B.6。

表 B.6 可追溯过程域的过程证据

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
需求	需求可跟踪性	软件制品到需求的可跟踪性等级	软件制品到需求的可跟踪性情况	等级型	2	2 部分软件需求在概要设计、详细设计、功能模块、代码等层次，建立可用的跟踪关系，有一定效果	同二级要求	3 对于关键的软件需求在概要设计、详细设计、功能模块、代码等层次，都建立了较为完整的跟踪关系，且有效	同四级要求
设计	设计可跟踪性	设计到需求的追踪程度	一个设计是否可追踪到对应的需求	布尔型	2	是	同二级要求	同二级要求	同二级要求
		需求到设计的追踪程度	一个需求是否可追踪到对应的设计	布尔型	2	是	同二级要求	同二级要求	同二级要求

表 B.6（续）

阶段	可信原则	过程证据名称	描述	类型	适用等级	二级	三级	四级	五级
编码	源代码可跟踪性	需求到原始需求和利益相关方的追踪程度	一个需求是否能够追踪回溯到原始需求和提出需求的利益相关方	布尔型	2	是	同二级要求	同二级要求	同二级要求
		需求到源代码的可追踪程度	一个需求是否可追踪到至少一块代码	布尔型	2	是	同二级要求	同二级要求	同二级要求
		源代码到设计的可追踪程度	一块代码是否可追踪到至少一个对应的设计	布尔型	2	是	同二级要求	同二级要求	同二级要求
		设计到源代码的可追踪程度	一个设计是否可追踪到实现它的源代码	布尔型	5	—	—	—	是
测试	测试可跟踪性	测试到需求的可追踪性	所用软件组件和配置项的测试对于需求是否是可跟踪的	布尔型	2	是	同二级要求	同二级要求	同二级要求
		测试到设计的可追踪性	测试是否可追踪到至少一个设计	布尔型	2	是	同二级要求	同二级要求	同二级要求
		测试到源代码的可追踪性	测试是否可追踪到至少一块代码	布尔型	2	是	同二级要求	同二级要求	同二级要求
注：“—”表明该证据在该等级不做要求。									

附 录 C
（规范性附录）
软件制品证据

C.1 功能性特性的制品证据

功能性质量特性共有 13 个证据，见表 C.1。

表 C.1 功能性特性的制品证据

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
设计	功能完备性	设计验证	是否对核心部件设计做过形式化等验证	$X=A/B$ A——设计满足的需求； B——所有需求	布尔型	4	—	N
测试	功能完备性	需求覆盖率	用来衡量测试覆盖到的需求情况	$X=A/B$ A——测试能追踪到的需求； B——所有需求	百分比	3	≥95%	N
测试	功能完备性	功能实现覆盖率	按照需求规约对系统做功能性测试。对在评价中检测到不能正确实现或缺失的功能数进行计数，将其与需求规约中描述的功能数相比较	$X=1-A/B$， A——在评价中测出的不能正确实现或缺少的功能数； B——需求规约中描述的功能数	百分比	3	≥70%	N
交付	功能稳定性	功能稳定性	系统投入运行后，对需要变更功能规约中描述的功能数进行计数，将其与需求规约中描述的功能总数相比较	$X=1-A/B$ A——在生存周期开发阶段变更的功能数； B——需求规约中描述的功能数	百分比	3	≥80%	N
软件开发全过程	功能性的依从性	功能性的依从性	依从现行标准或现有执行的标准中对功能性规定的程度	该依从性可分为三级：1)无；2)部分依从；3)完全依从	等级型	1	—	N
编码	功能依从性	代码中的注释率	该指标表示源代码中按照标准进行注释的程度。对含有注释的代码进行计数，并与所有代码行数进行比较	$X=A/B$ A——含有注释的代码行数； B——所有代码行数	百分比	1	≥10%	Y

表 C.1（续）

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
设计	功能正确性	设计的正确与完整性	客观地核实软件项目的实施行动与开发中产品遵从于对应的设计标准	$X=A/B$ A——满足需求的设计规约； B——所有设计规约	百分比	1	≥85%	N
编码	功能正确性	错误检测	对评审中检测到的故障进行计数，并与此阶段估计会检测出的故障数相比较	$X=A/B$ A——评审中发现的错误数； B——评审中估计的可能会发生的错误数	百分比	3	≥20%	N
测试	功能正确性	功能正确性	准确性需求已经完全实现的程度，对已经实现准确性需求的功能进行计数，并与有特定准确性需求的功能数相比较	$X=A/B$ A——在评价中已证实的正确实现的功能数； B——有特定准确性需求的功能数总数	百分比	3	≥99%	Y
测试	功能正确性	计算正确性	按照需求规约对系统做功能性测试。对在评价中已证实的实现准确性需求的功能计数，将其与需求规约中描述的功能数相比较	$X=A/B$ A——在评价的过程中已证实的实现准确性需求的功能数； B——需要实现特定准确性需求的功能数	百分比	3	≥99%	Y
测试	功能正确性	估算的缺陷密度	对在一定的试验周期内检测到的故障数进行计数，并用可靠性增长估计模型来预测未来潜在的故障数	$X=\lvert A_1-A_2\rvert/B$ X——评估残存的潜在故障密度； A_1——在软件产品中预测的潜在故障总数； A_2——实际已检测到的故障总数； B——产品规模	数值	5	≤2/KLOC	Y
测试	功能正确性	实际的缺陷密度	对检测到的故障个数进行计数并计算密度	$X=A/B$ A——检测出的缺陷数， B——产品规模	数值	3	≤3/KLOC（或遵循行业标准）	N
交付	功能正确性	接口一致性	对按规约已经正确实现的接口协议进行计数，并与规约中要实现的接口协议数相比较	$X=A/B$ A——在评审中已证实的按规约中正确实现的接口协议数； B——规约中要实现的接口协议数	百分比	3	≥99%	Y

注：“N”代表不可裁剪；“Y”代表可以根据需求进行裁剪；“—”代表该证据无阈值要求。

C.2 性能效率特性的制品证据

性能效率质量特性共有 12 个证据，见表 C.2。

表 C.2 性能效率特性的制品证据

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
交付	容量	事务处理能力	单位时间内能处理的事务数	$X=A/B$ A——在观察时间内所完成的事务数量； B——观察持续的时间(s)	数值	3	≥10 s	N
交付	容量	并发访问量	在特定时间内，能够同时访问系统的用户数是否达到要求	1 为达到要求； 0 为未达到要求	布尔型	3	—	N
交付	时间特性	平均响应时间	系统响应一个用户任务或系统任务的平均时间	$X=\mathrm{Sum}(A_i)/n$ A_i ——系统响应一个指定用户任务或系统任务在第 i 次测量所用的时间； n ——所测量的响应次数	数值	3	≤0.5 s～8 s 要考虑到客户端和服务端	N
交付	时间特性	响应时间的充分性	系统的响应时间与指定目标的满足程度	$X=A/B$ A——所测的平均响应时间， B——指定的目标响应时间	百分比	4	≥80%	Y
交付	时间特性	平均周转时间	完成一个工作或一个异步的过程的平均时间	$X=\mathrm{Sum}(B_i-A_i)/n$ A_i ——开始工作 i 的时间； B_i ——完成工作 i 的时间； n ——测量的数量	数值	4	≤10 s	Y
交付	时间特性	周转时间的充分性	周转时间满足指定目标的程度	$X=A/B$ A——测量的平均周转时间； B——指定的目标周转时间	百分比	4	≥80%	Y
交付	时间特性	平均吞吐率	单位时间执行的作业数量	$X=\mathrm{Sum}(A_i/B_i)/n$ A_i ——在第 i 次观察时间内完成的作业数量； B_i ——第 i 次观察时间长度； n ——观察的数量	数值	4	≥60	Y

表 C.2（续）

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
软件开发全过程	性能效率的依从性	性能效率的依从性	依从现行标准或现有执行的标准中对性能效率规定的程度	该依从性可分为三级： 1） 无； 2） 部分依从； 3） 完全依从	等级型	1	—	N
交付	资源利用性	平均处理器利用率	与操作时间相比，执行一组给定任务的处理时间	$X=\mathrm{Sum}(A_i/B_i)/n$ A_i ——在第 i 次观察中实际执行一组给定的任务的处理时间； B_i ——在第 i 次观察内执行任务的操作时间； n ——观察的次数	数值	4	≤0.7	N
交付	资源利用性	平均内存利用率	与可用的内存空间相比，被用于执行一组给定任务的内存比率	$X=\mathrm{Sum}(A_i/B_i)/n$ A_i ——在第 i 次样本处理中实际执行一组给定的任务的内存大小； B_i ——在第 i 次样本处理中可用的内存大小； n ——处理的样本数量	百分比	4	≤80%	Y
交付	资源利用性	平均I/O设备利用率	与I/O操作时间相比，执行一组给定任务所用I/O设备的忙碌时间所占的利率	$X=\mathrm{Sum}(A_i/B_i)/n$ A_i ——在第 i 次观察时输入/输出设备执行一组给定任务的持续时间； B_i ——在第 i 次观察室输入/输出操作执行任务的持续时间； n ——观察的数量	百分比	4	≤80%	Y
交付	资源利用性	带宽利用率	可用带宽中，被用于执行一组给定任务的比率	$X=A/B$ A ——在一段时间内执行一组给定任务的实际传输带宽； B ——执行一组给定任务的可用带宽容量	百分比	4	上行≤20%，下行≤60%	Y

注：“N”代表不可裁剪；“Y”代表可以根据需求进行裁剪；“—”代表该证据无阈值要求。

C.3 兼容性特性的制品证据

兼容性质量特性共有 5 个证据，见表 C.3。

表 C.3 兼容性特性的制品证据

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
设计	共存性	程序部件应用范围的广泛性	程序部件之间的共存性有多强，在不同环境中系统或程序其他组件变更的情况下，可不受影响而有共存的能力的部件比率	$X=A/B$ A ——不会受其他组件造成影响的组件数量； B ——共存的组件数量	百分比	2	≥70%	N
测试	共存性	环境或资源共享性	软件系统在公共环境与其共享资源的其他系统共存的能力	$X=A/B$ A ——不会对其他环境或资源造成影响的数量； B ——共存的环境或资源数量	百分比	2	≥85%	N
设计	互操作性	通信通用性	两个或两个以上的系统、程序组件或程序部件可交换信息，并使用已交换通信信息的能力	$X=A/B$ A ——不会对其他系统、程序组件或程序部件造成影响的数量； B ——互操作的系统、程序组件或程序部件数量	百分比	2	≥85%	N
测试	互操作性	数据互连性	连接一个软件和其他系统所需工作量的数量。如果这个软件要联网，或与其他系统通信，或要把其他系统纳入到自己的控制之下，应有系统间的接口，使之可以联结	$X=A/B$ A ——不会对其他互操作的软件或系统所需造成影响的数量； B ——互操作的所需的数量	百分比	2	≥95%	N
软件开发全过程	兼容性的依从性	兼容性的依从性	依从现行标准或现有执行的标准中对兼容性规定的程度	该依从性可分为三级： 1） 无； 2） 部分依从； 3） 完全依从	等级型	1	—	N

注：“N”代表不可裁剪；“—”代表该证据无阈值要求。

C.4 易用性特性的制品证据

易用性质量特性共有 12 个证据，见表 C.4。

表 C.4 易用性特性的制品证据

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
设计	可操作性	撤销能力	能够产生重大后果的任务有多少比例提供了再确认或者撤销的可选项	$X=A/B$ A ——提供撤销能力或者再确认提示的任务数量； B ——能够从提供撤销能力或者再确认提示中获益的任务数量	百分比	2	≥95%	N
测试	可操作性	信息明确性	系统信息多大程度上可向用户传达正确的结果或者指令	$X=A/B$ A ——可传达正确结果或者指令的系统信息数量； B ——系统信息的总数量	百分比	2	≥90%	Y
测试	可操作性	监控能力	在操作过程中，能够被监控功能的比例	$X=A/B$ A ——声明了监控能力的功能数量； B ——可从监控中获益的功能数量	百分比	3	≥80%	N
交付	易识别性	描述完整性	在产品说明或者用户文档中被描述的使用场景的比例	$X=A/B$ A ——在产品说明或者用户文档中被描述的使用场景数量； B ——产品的所有使用场景	百分比	3	≥90%	Y
交付	易识别性	示范覆盖率	拥有示范功能来帮助使用者判断适用性的任务的比例	$X=A/B$ A ——示范功能中展示的任务数量； B ——可能拥有示范功能的任务数量	百分比	3	≥90%	Y
交付	易识别性	站点自描述能力	当前站点登录页面的内容中，能解释该站点目的的登陆页面所占的比例	$X=A/B$ A ——能够解释站点目的的登录页面数量； B ——站点中的登录页面数量	百分比	3	≥90%	Y

表 C.4（续）

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
交付	易学性	用户指导完整性	在用户文档中或是帮助设施中有详细说明，以协助用户应用具体功能的功能比例	$X=A/B$ A ——用户文档或是帮助设施中描述的功能数量； B ——文档中要求的所有可执行的功能	百分比	3	≥80%	Y
交付	易学性	错误理解率	给出错误发生原因以及解决它的错误信息的比例	$X=A/B$ A ——能够解释错误原因和提供解决方案的错误声明数量； B ——系统的错误声明数量	百分比	4	≥70%	Y
软件开发全过程	易用性的依从性	易用性的依从性	依从现行标准或现有执行的标准中对易用性规定的程度	该依从性可分为三级： 1） 无； 2） 部分依从； 3） 完全依从	等级型	1	—	N
交付	易用性的描述	描述的完整和正确性	在产品支持文档中是否有对易用性的描述和解释	可根据需求包含下列证据的内容：描述完整性，示范覆盖率，站点自描述能力，用户指导完整性等	布尔型	2	—	N
测试	用户错误的防范机制	抵御误操作	可用来防止用户行为和输入产生系统故障的措施的比例	$X=A/B$ A ——可用来防止用户行为和输入产生系统故障的措施数量； B ——可设置的用来防止用户行为和输入产生系统故障的措施总数量	百分比	3	≥80%	N
交付	用户错误的防范机制	用户错误的修复	可被系统修正或者恢复的用户错误的比例	$X=A/B$ A ——在设计和测试过程中可被系统恢复的用户错误数量； B ——在运行过程中可能出现的用户错误数量	百分比	3	≥70%	Y

注：“N”代表不可裁剪；“Y”代表可以根据需求进行裁剪；“—”代表该证据无阈值要求。

C.5 可靠性特性的制品证据

可靠性质量特性共有 12 个证据，见表 C.5。

表 C.5 可靠性特性的制品证据

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
设计	成熟性	是否进行可靠性关键组件识别	是否根据对软件关键程度等级和可靠性关键组件进行识别	1 为已进行； 0 为未进行	布尔型	4	—	Y
设计	成熟性	是否进行关键组件可靠性分析	是否采用软件故障树分析、软件失效模式及影响分析、或其他软件可靠性评估模型进行可靠性分析	1 为已进行； 0 为未进行	布尔型	4	—	Y
测试	成熟性	测试覆盖度	包含在相关测试组件中的系统或软件功能中实际执行的百分比	$X=A/B$ A ——系统或软件功能实际执行的个数； B ——系统或软件功能包含在相关测试组件中的个数	百分比	2	≥95%	N
维护	成熟性	故障频率	在单位时间内平均的故障数目	$X=A/B$ A ——在观察期内检测到的故障个数； B ——观察的时间跨度	数值	3	≤3/100 h	Y
维护	成熟性	平均故障间隔时间	在系统/软件运行时的 MTBF 值	$X=A/B$ A ——运行时间； B ——系统/软件实际发生的故障个数	数值	3	≥200 h	Y
软件开发全过程	成熟性	缺陷修复	设计/编码/测试阶段检测到的与可靠性相关的缺陷中被修复的百分比	$X=A/B$ A ——被修复的与可靠性相关的缺陷个数； B ——检测到的与可靠性相关的缺陷个数	百分比	2	≥85%	N
软件开发全过程	可靠性的依从性	可靠性的依从性	依从现行标准或现有执行的标准中对可靠性规定的程度	该依从性可分为三级： 1） 无； 2） 部分依从； 3） 完全依从	等级型	1	—	N

表 C.5（续）

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
维护	可用性	平均宕机时间	一次故障发生时系统保持失效的时间	$X=A/B$ A ——总的宕机时间； B ——所观察到的中断次数	数值	3	≤5 h	Y
维护	可用性	系统可用性	规划的系统运行时间中实际可用的时间百分比	$X=A/B$ A ——系统实际运行时间； B ——规划的系统运行时间	百分比	3	≥95%	Y
设计	容错性	关键组件冗余度	对关键软件组件进行冗余设计的程度	4 级：故障容限 1、3 版本程序设计； 5 级：故障容限 2、5 版本程序设计	等级型	4	依据行业要求	Y
交付	易恢复性	平均恢复时间	软件/系统从故障中修复花费的时间	$X=\mathrm{Sum}(A_i)/n$ A_i ——软件/系统在每次故障中的恢复时间总数； n ——故障个数	数值	3	≤5 h	Y
交付	易恢复性	数据备份的完成度	通常情况下备份的数据项占比	$X=A/B$ A ——实际备份的数据项个数； B ——需要用作故障恢复的数据项个数	百分比	3	≥80%	N
注：“N”代表不可裁剪；“Y”代表可以根据需求进行裁剪；“—”代表该证据无阈值要求。								

C.6 信息安全性特性的制品证据

信息安全性质量特性共有 12 个证据，见表 C.6。

表 C.6 信息安全性特性的制品证据

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
设计	保密性	数据加密正确性	按照规约执行的数据项加密/解密的正确性	$X=A/B$ A ——按规约已正确加密/解密的数据项个数； B ——规约中要求加密/解密的数据项个数	百分比	3	≥90%	Y
设计	保密性	密码算法的强度	密码算法被仔细审查的程度，由专家审查给出的密码算法强度的判断	$X=1-A/B$ A ——专家审查后，被攻破或者存在不可接受的风险的密码算法的个数； B ——使用的密码算法的个数	百分比	4	≥95%	N

表 C.6（续）

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
测试	保密性	访问可控性	按照规约，在未授权的访问中被保护的加密（机密的）数据项的百分比	$X=1-A/B$ A ——未授权的情况下可访问的机密数据个数； B ——要求访问控制的数据项个数	百分比	3	≥40%	N
设计	抗抵赖性	唯一、特定身份鉴别方式的使用	是否采用数字签名的处理需要抗抵赖的事件	是否使用了唯一、特定身份认证方式	布尔型	3	—	Y
设计	可核查性	用户审计跟踪的完整性	与用户访问系统或者数据相关的审计跟踪的完整度	$X=A/B$ A ——在日志中记录的访问个数； B ——实际测试过的访问系统或者数据的个数	百分比	3	≥60%	Y
设计	可核查性	系统日志的保留	在规定的保存期限内，系统日志保存在存储器中的时间占比	$X=A/B$ A ——系统日志实际保存在存储器中的时间区间； B ——系统日志在存储器中保存的规定期限（天数）	百分比	3	≥99%	N
交付	完整性	数据被破坏性	在未授权的访问中被破坏或者修改的数据项百分比	$X=A/B$ A ——在未授权的访问中实际被破坏的数据项个数； B ——需要避免被修改或者破坏的数据项个数	百分比	3	≤10%	Y
交付	完整性	数据破坏保护	是否有合适的数据保护方法	是否有合适的数据保护方法	布尔型	3	—	N
交付	完整性	数据完整性保护	检查中，因内存溢出、传送、存储等原因被破坏的测试用例的个数占应检查的测试用例个数的比率	$X=A/B$ A ——因内存溢出、传送、存储等原因被破坏的测试用例的个数； B ——应检查的测试用例个数	百分比	3	≤10%	Y
软件开发全过程	信息安全性的依从性	信息安全性的依从性	依从现行标准或现有执行的标准中对信息安全性规定的程度	该依从性可分为三级： 1） 无； 2） 部分依从； 3） 完全依从	等级型	1	—	N

表 C.6（续）

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
测试	真实性	验证规则的一致性	执行的验证规则的百分比	$X=A/B$ A ——执行的验证规则的个数； B ——要求的验证规则的个数	百分比	4	≥80%	N
交付	真实性	验证机制的充分性	系统能够验证主体身份的程度	$X=A/B$ A ——提供的验证机制的个数（比如用户 ID/密码）； B ——规定的验证机制的个数	百分比	3	≥80%	N
注：“N”代表不可裁剪；“Y”代表可以根据需求进行裁剪；“—”代表该证据无阈值要求。								

C.7 维护性特性的制品证据

维护性质量特性共有 12 个证据，见表 C.7。

表 C.7 维护性特性的制品证据

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
软件开发全过程	维护性的依从性	维护性的依从性	依从现行标准或现有执行的标准中对可维护性规定的程度	该依从性可分为三级： 1） 无； 2） 部分依从； 3） 完全依从	等级型	1	—	N
测试	可重用性	资产可重用性	系统中，可重用资产的比例	$X=A/B$ A ——已重用的设计和实施的资产数量； B ——系统中的资产数量	百分比	3	≥50%	Y
测试	可重用性	编码规则一致性	遵循规定编码规则的模块的比例	$X=A/B$ A ——符合特定系统编码规则的软件模块数量； B ——已实施的软件模块数量	百分比	2	≥80%	N
设计	模块化	组件耦合度	组件之间的独立性有多强，可不受系统或程序中其他组件变更的影响（耦合度测量的等价值）的组件的比例	$X=A/B$ A ——已实施的不会对其他组件造成影响的组件数量； B ——需要独立的组件数量	百分比	4	≥70%	Y

表 C.7（续）

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
测试	模块化	充分的圈复杂度	具有可接受的圈复杂度的软件模块的比例	$X=1-A/B$ A ——圈复杂度分值超过指定阈值的软件模块数量； B ——已实施的软件模块数量	数值	4	≤15	Y
测试	易测试性	测试功能的完整性	已执行测试功能和工具的完整程度	$X=A/B$ A ——按照说明执行的测试功能数量； B ——要求的测试功能数量	百分比	3	≥90%	N
测试	易测试性	测试的自主性	软件被测试的独立程度	$X=A/B$ A ——依赖其他系统的测试中能够被存根模拟的测试数量； B ——依赖其他系统的测试数量	百分比	4	≥80%	Y
测试	易测试性	测试的可重启性	维护后能在重启点执行测试操作的容易程度	$X=A/B$ A ——执行测试过程中在逐步检查的请求点上维护人员可暂停和重启的用例数量； B ——执行测试过程可被暂停的用例数量	百分比	4	≥80%	Y
测试	易分析性	诊断功能的有效性	满足因果分析要求的诊断功能的比例	$X=A/B$ A ——用于因果分析的诊断功能数量； B ——已实施的诊断功能数量	百分比	4	≥50%	Y
测试	易分析性	诊断功能的充分性	已经实施必要的诊断功能的比例	$X=A/B$ A ——已实施的诊断功能数量； B ——要求的诊断功能数量	百分比	3	≥80%	Y
测试	易修改性	修改的效率	和预期时间相比，修改的效率	$X=\mathrm{Sum}(A_i/B_i)/n$ A_i ——进行特定修改所花费的总工时； B_i ——进行特定修改预期的时间； n ——修改的数量	数值	3	≤1	N
交付	易修改性	修改的正确性	修改能被正确执行的比例	$X=1-A/B$ A ——实施后在已定义时间内造成事故或失败的修改数量； B ——已实施的修改数量	百分比	4	≥80%	N
注：“N”代表不可裁剪；“Y”代表可以根据需求进行裁剪；“—”代表该证据无阈值要求。								

C.8 可移植性特性的制品证据

可移植性质量特性共有 6 个证据，见表 C.8。

表 C.8 可移植性特性的制品证据

阶段	子特性	制品证据名称	描述	计算方式	类型	适用等级	需要满足的阈值	可裁剪
软件开发全过程	可移植性的依从性	可移植性的依从性	依从现行标准或现有执行的标准中对可移植性规定的程度	该依从性可分为三级： 1） 无； 2） 部分依从； 3） 完全依从	等级型	1	—	N
交付	适应性	系统软件环境的适应性	软件或者系统中，能够适应不同的系统软件环境的功能的比例	$X=1-A/B$ A ——经测试不能执行或者结果不能充分满足需求的功能个数； B ——在不同系统软件环境下测试的功能个数	百分比	3	≥90%	N
交付	适应性	硬件环境的适应性	软件或者系统中，能够适应不同的硬件环境的功能的比例	$X=1-A/B$ A ——经测试不能执行或者结果不能充分满足需求的功能个数； B ——在不同硬件环境下测试的功能个数	百分比	4	≥90%	N
交付	适应性	操作环境的适应性	软件或者系统中，能够适应特殊的操作和运行环境的功能的比例	$X=1-A/B$ A ——在运行测试中，不能执行或者结果不能充分满足需求的功能个数； B——在不同操作/运行环境下测试的功能个数	百分比	5	≥90%	N
交付	易安装性	易安装性/安装的灵活性	用户或维护者能否根据自己习惯定制安装进程的实例比例	$X=A/B$ A ——用户成功定制安装进程的实例数； B ——用户尝试定制安装进程的实例数	百分比	2	≥80%	N
交付	易安装性	安装时间的效率	实际安装时间相比预期时间的效率	$X=\mathrm{Sum}(A_i/B_i)/n$ A_i ——花费在安装 i 上的所有工作时间； B_i ——预期安装 i 所需的时间； n ——安装个数	百分比	2	≤95%	Y

注：“N”代表不可裁剪；“Y”代表可以根据需求进行裁剪；“—”代表该证据无阈值要求。

附 录 D
（资料性附录）
软件可信度评估示例

D.1 软件过程可信度评估

D.1.1 过程证据的可信度评估项目背景

某项目软件过程可信度等级要求为可信四级。下面对该项目的软件过程的实际可信度等级进行评估。

D.1.2 过程证据的可信度评估

收集数据，并对133个过程证据的可信等级进行评估。因为该项目要求的可信等级是四级，所以只需要看四级及以下指标。根据表5，可看出四级及以下指标共有130个。

表D.1给出了“可追溯”可信过程域下过程证据情况的例子。可看出：1)“可追溯”可信过程域一共有10个可信证据，其中需要关注的是除可信证据7以外的9个过程可信证据；2)9个可信证据中有一个可信证据没有达到可信四级要求。证据1是评估的弱项，需要重点关注和提升。其余可信证据也和该可信过程域证据可信评估的评估方法一样。

表D.1 “可追溯”可信过程域下过程证据的可信情况

序号	名称	类型	适用等级	最大证据等级	取值	实际可达到的可信等级	对目标可信等级的满足情况
1	软件制品到需求的可跟踪性等级	等级型	2	4	2	3	不满足
2	设计到需求的追踪程度	布尔型	2	2	Y	2	满足
3	需求到设计的追踪程度	布尔型	2	2	Y	2	满足
4	需求到原始需求和利益相关方的追踪程度	布尔型	2	2	Y	2	满足
5	需求到源代码的可追踪程度	布尔型	2	2	Y	2	满足
6	源代码到设计的可追踪程度	布尔型	2	2	Y	2	满足
7	设计到源代码的可追踪程度	布尔型	5	5	—	—	—
8	测试到需求的可追踪性	布尔型	2	2	Y	2	满足
9	测试到设计的可追踪性	布尔型	2	2	Y	2	满足
10	测试到源代码的可追踪性	布尔型	2	2	Y	2	满足
注：“—”代表该证据不考虑。							

D.1.3 可信原则的可信度判定

可信原则的可信度判定，可依据其对应的证据的可信情况。详细信息见表D.2。可看出可信原则“需求可追踪性”不满足目标可信要求。其余可信原则也和该过程下的可信原则可信判定相同。

表 D.2 “可追溯”可信过程域下可信原则的可信情况

序号	可信原则	对应证据(表 D.1)	实际可达到的可信等级	对目标可信等级的满足情况
1	需求可跟踪性	1	3	不满足
2	设计可追踪性	2,3	2	满足
3	源代码可追踪性	4,5,6,7	1	满足
4	测试可追踪性	8,9,10	2	满足

D.1.4 软件过程的可信评估

软件过程的可信度评估遵循木桶原理,为评估某项目的整个过程的可信等级,可得出在所有适用的可信原则中,可信等级最小的可信原则满足可信三级,即该项目满足可信三级,没有达到预定的可信四级等级要求。

图 D.1 给出了 133 个可信证据的情况,其中横坐标表示按照可信过程域分类的所有可信原则,纵坐标表示每个级别下的可信证据个数。图中不同颜色的方块表示该可信证据的可信等级不同,其中标记有黑点的方块表示取值已到该可信证据等级的最大值。对应不同的可信要求,如果其级别及其以下级别的所有证据的色块中没有低于其要求级别的色块,或者若有低级的色块,但有黑色圆点表明已到最高级别,则说明该级别的可信要求已经满足。图 D.1 证据视图中二级指标中没有白色色块,三级中没有三级以下色块,四级中有白色色块并且四级以下有三级色块,说明过程可信证据满足可信三级的要求,但没有达到预定的可信四级等级要求。

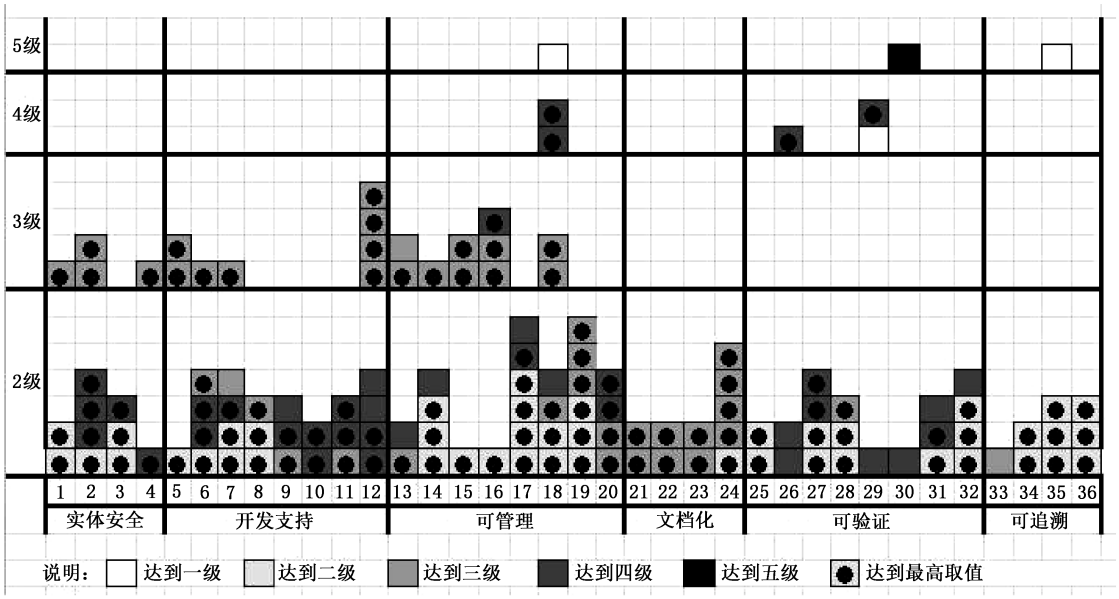

图 D.1 某项目过程证据可信情况

D.2 软件制品可信度评估

D.2.1 制品证据的可信度评估项目背景

某项目软件制品可信度等级要求为可信四级。下面对该项目的软件制品的实际可信度等级进行评估。

D.2.2 制品证据的可信度评估

收集数据，并对 84 个制品证据的可信等级进行评估。因为该项目要求的可信等级是四级，所以只需要看四级及以下指标。根据表 7，可看出四级及以下指标共有 82 个。

表 D.3 给出了“可靠性”质量特性下制品证据情况的例子。可看出：

a) “可靠性”质量特性下一共 12 个可信证据。证据 1 和 9 与现行标准冲突，因为依从现行标准，所以裁剪掉 1 和 9，因此只需要关注其他 10 个可信证据。

b) 10 个可信证据中有一个可信证据没有达到可信四级要求。证据 2 是评估的弱项，需要重点关注和提升。其余可信证据也和该可信证据可信评估的评估方法一样。

表 D.3 “兼容性”质量特性下制品证据的可信情况

序号	名称	类型	适用等级	最大证据等级	是否可裁剪	是否满足	对目标可信等级的满足情况
1	是否进行可靠性关键组件识别	布尔型	4	—	是	(—)已裁剪	—
2	是否进行关键组件可靠性分析	布尔型	4	—	是	否	不满足
3	测试覆盖度	百分比	2	≥95%	否	是	满足
4	故障频率	数值	3	≤3/100 h	否	是	满足
5	平均故障间隔时间	数值	3	≥200 h	否	1	满足
6	缺陷修复	百分比	2	≥85%	否	是	满足
7	行业标准对可靠性的依从性	等级型	1	—	否	2	满足
8	平均宕机时间	数值	3	≤5 h	是	是	满足
9	系统可用性	百分比	3	≥95%	是	(—)已裁剪	—
10	关键组件冗余度	等级型	4	依据行业要求	是	是	满足
11	平均恢复时间	数值	3	≤5 h	否	是	满足
12	数据备份的完成度	百分比	3	≥80%	否	是	满足
注：“—”代表该证据不考虑。							

D.2.3 软件制品的可信度评估

软件制品的可信度评估遵循木桶原理，为评估某项目的整个制品的可信等级，可得出在所有适用的证据中，可信等级最小的可信原则满足可信三级，即该项目满足可信三级，没有达到预定的可信四级等级要求。

图 D.2 给出了 84 个证据的情况，其中横坐标表示按照质量特性分类的 37 个子特性，纵坐标表示每个级别下的证据个数。图中不同颜色的方块表示该证据的可信等级不同，其中标记有“×”的方块表示裁剪的证据。对应不同的可信要求，如果其级别及其以下级别的所有证据的色块中没有低于其要求级别的色块，或者若有低级的色块，但有“×”表明已裁剪，则说明该级别的可信要求已经满足。图 D.2 证据视图中四级及四级以下证据中，只在四级上有一个不满足的制品证据，因此说明制品证据满足可信三级的要求，但没有达到预定的可信四级等级要求。

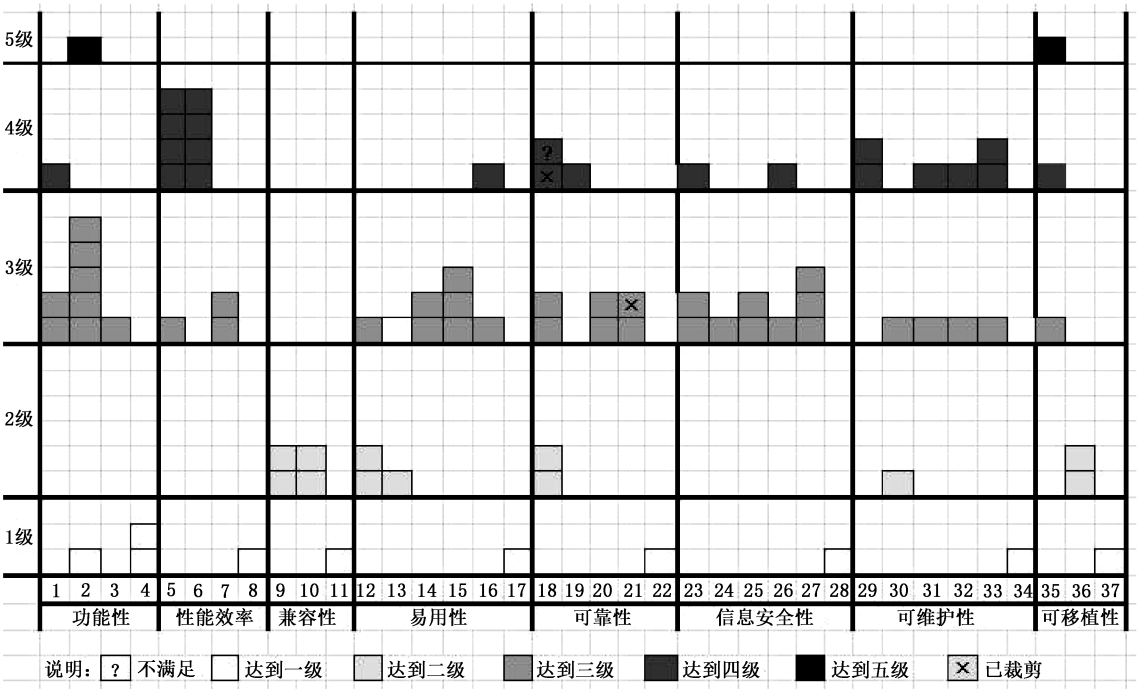

图 D.2　某项目制品证据可信情况

参 考 文 献

[1] GB/T 8566—2007 信息技术 软件生存周期过程
[2] GB/T 8567—2006 计算机软件文档编制规范
[3] GB/T 11457—2006 信息技术 软件工程术语
[4] GB/T 14394—2008 计算机软件可靠性和可维护性管理
[5] GB/T 16680—2015 系统与软件工程 用户文档的管理者要求
[6] GB/T 19000—2016 质量管理体系 基础和术语
[7] GB/T 20158—2016 信息技术 软件生存周期过程 配置管理
[8] GB/T 20271—2006 信息安全技术 信息系统通用安全技术要求
[9] GB/T 20272—2006 信息安全技术 操作系统安全技术要求
[10] GB/T 20917—2007 软件工程 软件测量过程
[11] GB/T 22239—2008 信息安全技术 信息系统安全等级保护基本要求
[12] GB/T 25000.1—2010 软件工程 软件产品质量要求与评价(SQuaRE)SQuaRE 指南
[13] GB/T 29829—2013 信息安全技术 可信计算密码支撑平台功能与接口规范
[14] GB/Z 31102—2014 软件工程 软件工程知识体系指南
[15] SJ/T 11234—2001 软件过程能力评估模型
[16] SJ/T 11235—2001 软件能力成熟度模型
[17] ISO/IEC 25023:2016 Systems and software engineering—Systems and software Quality Requirements and Evaluation(SQuaRE)—Measurement of system and software product quality
[18] CMMI-DEV V1.3,2010
[19] CMMI-DEV V2.0,2018