软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(7):1713−1731 [doi: 10.13328/j.cnki.jos.005102]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于证据的软件过程可信度模型及评估方法*

王德鑫[1,2,3], 王 青[1,2,3], 贺 劼[4]

[1](中国科学院 软件研究所 互联网软件技术实验室,北京 100190)

[2](中国科学院大学,北京 100190)

[3](计算机科学国家重点实验室(中国科学院 软件研究所),北京 100190)

[4](中国科学院 软件研究所 天基综合信息系统重点实验室,北京 100190)

通讯作者: 王德鑫, E-mail: wangdexin@itechs.iscas.ac.cn 王青, E-mail: wq@itechs.iscas.ac.cn

摘 要: 软件可信已经是一个迫在眉睫的重要问题,但对软件可信性的评估却一直没有一个系统且客观的标准.一些研究工作从可信证据的采集渠道入手,譬如认为有第 3 方测试的证据,其可信级别就高一些,而若有用户的使用反馈则可信级别就更高.这些工作在可信的客观性方面做了很好的贡献.但可信其实是一个系统性的问题,而且质量形成于过程,其证据的充分必要程度以及对必要开发过程的覆盖程度等非常关键.基于软件开发过程,从过程的实体、行为以及制品 3 个方面提取软件可信的证据,建立了由 37 个可信原则、182 个过程可信证据和 108 个制品可信程度证据组成的软件过程可信度模型,并给出基于该模型证据的软件过程可信评估方法,试图从开发过程的可信程度来建立软件产品的可信的信心.

关键词: 软件可信;过程可信;软件制品可信

中图法分类号: TP311

中文引用格式: 王德鑫,王青,贺劼.基于证据的软件过程可信度模型及评估方法.软件学报,2017,28(7):1713−1731. http://www.jos.org.cn/1000-9825/5102.htm

英文引用格式: Wang DX, Wang Q, He J. Evidence-Based software process trustworthiness model and evaluation method. Ruan Jian Xue Bao/Journal of Software, 2017,28(7):1713−1731 (in Chinese). http://www.jos.org.cn/1000-9825/5102.htm

Evidence-Based Software Process Trustworthiness Model and Evaluation Method

WANG De-Xin[1,2,3], WANG Qing[1,2,3], HE Jie[4]

[1](Laboratory for Internet Software Technologies, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](University of Chinese Academy of Sciences, Beijing 100190, China)

[3](State Key Laboratory of Computer Science (Institute of Software, The Chinese Academy of Sciences), Beijing 100190, China)

[4](Science & Technology on Integrated Information System Laboratory, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Today's software is required to be more trustworthy due to its ever more important role in the society. However there is still lack of systematic and objective criteria for the evaluation of software trustworthiness. Existing research focuses on how to get the evidence, with the assumption that system is more trustworthy if the evidence is obtained from a third party test, or from the feedback of past users. Although such study contributes to the objectivity of trustworthiness, the process-oriented nature of system trust is not well addressed. In this case, the sufficiency and necessity of software process related evidence, as well as the coverage ratio of the necessary development process, are critical. This paper attempts to establish the confidence of software product from the trustworthiness of

* 基金项目: 国家自然科学基金(91318301, 91218302)
Foundation item: National Natural Science Foundation of China (91318301, 91218302)
收稿时间: 2016-01-21; 修改时间: 2016-03-25; 采用时间: 2016-05-17; jos 在线出版时间: 2016-07-30
CNKI 网络优先出版: 2016-08-01 09:38:55, http://www.cnki.net/kcms/detail/11.2560.TP.20160801.0938.004.html

development process. Based on the software development process, software trustworthiness is determined by three aspects: process entity, behavior and products. A software process trustworthiness model is proposed that includes 37 trustworthiness principles, 182 process entities and behaviors evidences, and 108 artifacts evidences. Based on this model, an evaluation method for process trustworthiness is also developed.

Key words: software trustworthiness; process trustworthiness; software artifacts trustworthiness

1 引 言

信息时代下,软件的应用领域不断扩展,在国民经济中发挥越来越重要的作用.软件不仅应用于航空航天、能源电信等关乎国家命脉的工业领域,还几乎覆盖了日常社会生活的各个领域.软件的需求和应用越来越多,而同时人们对软件质量的容忍度却越来越低,这也导致软件的可信问题越来越突出.随着软件的复杂度越来越高,软件失效的现象时有发生,给人们的工作和生活带来不便,严重的甚至会影响国家安全,软件产品并不总是可以被信任的,有时会产生难以预期的结果,Boehm[1]曾提出,提高软件的可信性已成为未来软件发展的重要方向之一.

目前对于软件可信性(software trustworthiness)的理解有很多种,一种比较共同的认识是把"软件可信"定义为一系列软件质量属性的证据集合以及这些证据集合满足需求的置信度或可信度[2−4],这些可信证据的特征与缺陷、测试、形式化验证方法、信息安全等技术密切相关[4,5].

现代质量管理理论强调,通过过程控制质量属性的实现,美国国家安全局(National Security Agency,简称NSA)联合其他政府和企业组织提出可信软件方法学(TSM)[6],希望通过评估软件开发过程的行为和能力来保证其开发软件的可信,但在 TSM 中只定义了过程可信的基本原则,需要评估人员依赖于自己的主观认识和经验,解释看到的现象,并给出评估的结果.这使得评估结果的客观性和可比性受到一定的影响.

本文参考 TSM 的 44 个可信原则,从现代质量管理的基本元素出发,提出了可信实体、可信行为、可信制品 3 个可信保障目标,结合软件过程的实践活动特点,建立了 6 类并覆盖整个软件过程生命周期的 36 个过程可信原则以及 1 个覆盖软件生命周期的制品可信原则.针对每一个可信原则,我们还开发了支持其分级评估的可信证据集合.本文工作首次提出了一个基于软件开发过程中数据所形成的证据,量化评估软件过程和过程制品可信程度的模型,可以有效地支持软件过程可信评估的客观性和全面性.

本文第 2 节讨论相关研究.第 3 节介绍本文重点提出的过程可信度模型,包括构建证据模型的过程、模型介绍和 37 个可信原则的组成.第 4 节在可信模型的基础上详细阐述过程可信证据和制品可信证据,包括证据的组成和等级分布.第 5 节提出软件过程可信评估方法.第 6 节通过几个实际案例展示和验证本方法的适用性.第 7 节讨论和总结本研究的结论,并对未来研究工作加以展望.

2 相关研究

软件可信的研究主要涉及软件的功能完整性、安全性、可维护性、可靠性等,研究者从不同的视角,解释可信的不同刻面,并利用特定领域条件下对软件产品的测试、度量和评价技术,甚至形式化方法和模型验证技术来提供软件可信的证据.

关于软件可信的定义有很多,比如可信计算机国家评估标准(TCSEC)[7]把安全性作为软件可信的唯一标准;Parnas[8]认为软件可信是利用软件工程技术减少软件失效的能力,包括增强测试、回访和检测技术;而通用标准(CC)[9]提出评价软件安全性的完整框架,并将其作为可信评价的标准来执行.

现代质量理论认为,质量形成于过程,可信作为一种对质量的要求,也必然不可能脱离过程而孤立存在.一些研究者提出面向过程的可信管理方法,来保障交付产品的可信性.比如可信软件方法学(TSM)[2,6],TSM 是美国国家安全局和其他 3 家机构共同提出的,根据软件过程的特点提出 44 个可信原则来定义软件可信,但并未提出支撑可信原则评估的证据,所以 TSM 虽然提供了评估软件可信性的指导方法,但在 TSM 可信评估中,评估人员往往根据个人经验对 44 个可信原则的满足情况进行定性判断,评估人员的主观因素容易影响可信评估结果的客观性和准确性.

国内近年也开展了大量软件可信保障方面的研究,国家自然科学基金委专门部署了软件可信研究的重大计划,国家 863 计划也部署可信软件开发环境方面的重点课题.很多研究者开展了大量相关研究.譬如陈火旺[3]认为软件系统的可信性质是指该系统需要满足的关键性质;当软件一旦违背这些关键性质就会造成不可容忍的损失时,称这些关键性质为高可信(high confidence)性质,同时他还强调了形式化方法、需求分析、设计和测试技术以及过程技术在开发和保障高可信软件系统中的重要作用.蔡斯博[10]提出了一种支持软件资源可信评估的框架,该框架中包括证据收集、证据信任管理和可信评估等技术.Tan[11]针对软件可信性度量进行了基于属性的研究,他认为软件的行为及其产生的结果可以用一组属性来表示,软件的可信性也可以通过一组属性以及用户在这组属性上的预期来共同定义.刘旭东[12]提出软件过程可信度框架,认为过程可信可以作为产品可信的重要指证.陈仪香[13]提出了一种软件可信的度量模型及分级评价方法,通过航天软件的可信属性来研究软件的可信度和定量分析评价,该航天嵌入式软件可信性度量模型及分级评价方法能够有效地评价航天嵌入式软件的可信性并发现软件产品研制过程中需要加强的部分.

此外,CMMI[14]、SEE-CMM[15]以及国际标准 ISO 9001[16]和 9126[17]等,虽然没有专门提出软件可信的概念,但作为业界最为广泛采用的过程方法和标准,为基于过程的软件可信保障研究创造了良好的基础.其中,CMMI 是被业界广泛采用的软件过程管理框架,定义了过程管理、项目管理、支持过程和工程过程 4 类共 22 个软件过程域,在每个过程域中定义了一组过程实践来支持过程域的实现,并强调通过持续的过程改进来提高产品的质量.ISO 9001 是工业界采用最为广泛的质量管理标准,强调基于过程的质量管理.ISO 9126 则是软件产品的质量标准,定义了 6 类质量属性.

本研究小组从 2005 年开始软件质量管理和保障技术的研究,本文提出的模型希望从软件过程实体、行为、制品可信的角度,保障最终产品的可信程度.旨在将目前软件过程技术的研究从过程的可管理性提高到管理的可信性.我们提出了一个基于过程证据的软件过程可信度模型,其中研究并提出了 36 个面向过程实体和行为的可信原则,包括 182 个可信证据;1 个制品可信原则,包括 108 个可信证据.而且本文提出的可信度模型是一个开放的模型,不局限于某个特定的应用领域,软件项目和组织可以根据具体的可信要求裁剪和增加合适的证据子集.基于模型证据,本文还提出了可以量化的软件过程可信度评估方法,用于指导软件过程的可信管理和改进.

3 软件过程可信度模型

质量形成于过程,建立可信过程的目的在于通过过程实体、行为和结果的可信,满足最终产品的可信.我们把“过程可信度”定义为软件过程可生产满足期望的产品的信心度.用一组可信原则的满足性来衡量,如下表示.

定义 1. $SPTModel=\langle TPSet,Satisfied,Required\text{-}Level\rangle,TPSet=\{TP_1,TP_2,\ldots,TP_{37}\}$.

其中,*TP* 表示一个可信原则,是构成本模型的基本成分.*Required-Level* 表示软件项目需要满足的可信等级.*Satisfied* 表示整个软件过程的可信度是否满足 *Required-Level* 的要求,True 表示满足,False 表示不满足.

可信原则表示可信的通用要求,依据过程管理人、机、料、法、环 5 要素,我们以可信保障目标为导向,确定了 6 类可信原则,每类称为一个可信过程域.每个原则由不同的过程实践活动来实现.这些实践活动被实现的程度代表了其行为或者结果可信的程度.譬如:是否选择了合适的开发工具和方法,工作环境适应开发要求的程度,评审的效力和符合性等.这些实践或者属于软件生命周期的某个阶段的活动,或者是共性支撑活动,跨越整个生命周期.如图 1 所示.

每个可信原则由一组其所关心的活动产生的证据支持,如下表示.

定义 2. $TP=\langle ESet,Phase,ProcessArea,S\text{-}Level\rangle,ESet=\{Evident_1,Evident_2,\ldots,Evident_n\}$.

其中,*ESet* 表示该原则所属的证据集合,具体的关于证据的相关内容在第 4 节中介绍.*S-Level* 表示该可信原则的可信等级.*Phase* 表示该原则所属的阶段.$Phase\in\{Environment,Requirement,Designing,Coding,Testing,Life\ cycle\}$.

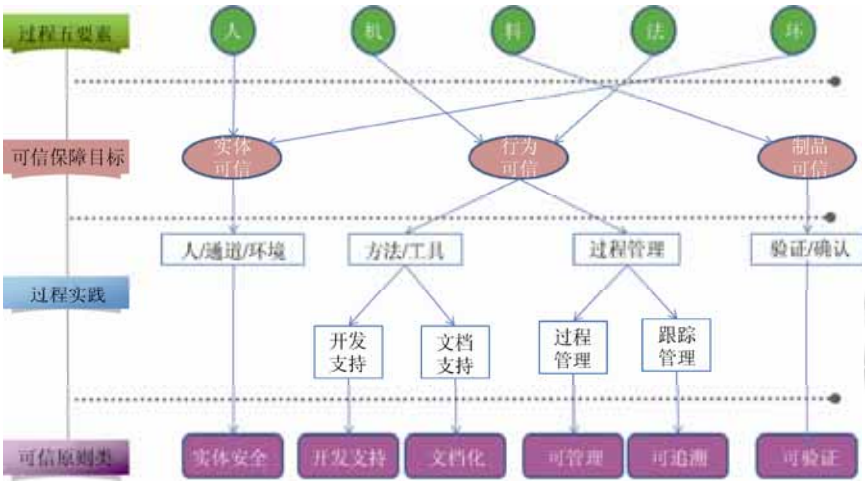

Fig.1 Software process trustworthiness principles and guarantee goals

图 1 软件过程可信原则及保障目标

Environment 表示软件开发所需要建立的软件工程环境,通常是软件开发的准备阶段,也有可能随开发的进行而加以补充和更新,从而贯穿整个开发周期;*Requirement*、*Designing*、*Coding*、*Testing* 表示软件开发周期中的需求、设计、源代码、测试阶段;*Life cycle* 表示软件开发过程的全生命周期.

ProcessArea 表示可信原则对应的关键活动属性,称为可信过程域.

ProcessArea∈{*Software development support*,*Documentation*,*V&V*,*Management*,*Traceability*,*Entities security*}.

Software development support 即开发支持,表示软件开发所需的工具方法,以及对重用和开源软件的支持等要求;*Documentation* 即文档化,表示软件开发过程各个阶段涉及的文档编制和实践活动要求;*V&V* 即可验证,关注软件开发过程各个阶段与软件制品的验证与确认相关的实践活动要求;*Management* 即可管理,表示软件开发管理涉及到的开发计划、风险控制和审计相关的实践活动要求;*Traceability* 即可追溯,表示对软件制品的追踪管理要求;*Entities security* 即实体安全,表示开发过程中对实体、环境、过程等信息安全的可信保障要求.

本模型共建立了 37 个可信原则,其中 1 个是制品可信原则,贯穿软件开发的全生命周期.其余 36 个分别分解在 6 类可信过程域和生命周期阶段中.如图 2 所示.

类别	环境	需求	设计	源代码	测试	全周期	合计
实体安全						方针(安全方针) 策略(多人控制，身份验证 最小特权) 工具(权限控制) 环境(可信信道，入侵检验 可信分配)	4
可管理	管理方针 环境完整性 知识共享					计划 风险管理 度量与分析 配置管理 过程审计	8
可追溯		需求可追踪性	设计可追踪性	源码可追踪性	测试可追踪性		4
开发支持	开发环境工具 重用支持 开源支持	工具	工具	源码编写标准 源码分析	工具		8
文档化		需求文档化	设计文档化	源码文档化	测试文档化		4
可验证	形式化验证要求	需求评审 形式化验证	设计评审 形式化验证	源码评审 形式化验证	测试评审		8
合计	7	5	5	6	4	9	36

Fig.2 Classifying organization view of trustworthiness principles

图 2 可信原则分类组织视图

3.1 开发支持

开发支持可信过程域共有 8 个可信原则,涉及软件开发环境建设、需求分析与管理、软件设计、代码开发、测试 5 个阶段.

1) 软件开发环境:共有 3 个软件开发支持方面的可信原则,分别是:

● 开发环境工具.该原则要求软件工程环境和所有的软件工具应当根据一个明确的选择策略来进行选择,选择策略中应当考虑可信等级、成熟度、文档和源码的可获取性等因素.代表证据如“环境和工具的培训计划”,即是否针对开发软件系统选择的环境和工具制定培训计划,证据类型是布尔型,可信等级 2 级及以上为 1.

● 重用支持.该原则表示是否要构建可重用的组件以及可重用的程度.代表证据如“软件的可重用程度”,该可信证据描述为“重用需求的等级是由最终软件将被重用的范围决定的:(1) Low(没有重用需求);(2) Normal(软件模块在项目范围内被重用);(3) High(软件模块将会在本组织的多个项目中被重用);(4) Very High(软件模块将会在多个组织的多个项目中被重用);(5) Extra High(软件模块可在多领域、多组织的不同项目中被重用,重用性是开发组织的主要目标)”,证据类型是等级类型,可信 2 级需要满足 Normal 的要求,3 级要满足 High 的要求,而 4 级及 4 级以上可信等级则要达到 Very High.

● 开源支持.开源软件应该接受选择,清晰的选择政策应该考虑可信等级、成熟度、文档和可获取的源码.代表证据如“开源软件质量”,该可信证据描述为“开源软件质量分为 3 等:(1) 低(未通过系统测试或未测试的代码);(2) 中(通过了系统测试的开源软件);(3) 高(公开发布或被实践验证的软件产品),是一个等级类型的可信证据,可信 2 级需要满足质量等级 2 级的要求,可信 3 级及 3 级以上需要满足质量等级 3 级的要求.

2) 需求分析与管理:该阶段对应软件开发过程的需求规约、需求分析和需求跟踪,主要目标是保证软件需求的完整性、一致性和正确性.在软件开发支持域有 1 个可信原则.

需求分析工具.使用需求分析 CASE 工具以支持需求规约、一致性检查和文档生成.代表证据如“软件需求分析自动化环境”,该可信证据描述为“表示为使用自动化工具管理软件需求的程度分级:(1) 未使用 CASE 工具管理需求;(2) 使用CASE工具覆盖少量需求管理活动,效果不明显;(3) 使用CASE工具覆盖部分需求管理活动,有一定效果;(4) 使用 CASE 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,非常有效;(5) 使用 CASE 工具全面覆盖需求规约、一致性检查、需求跟踪和文档生成,非常有效”,等级类型的可信证据,可信 2 级需要满足程度分级 2 级的要求,可信级别越高则需要的程度分级也逐级提高.

3) 软件设计:该阶段的主要目标是在满足软件需求的条件下,保证软件设计的完整性、一致性和正确性,并实现系统设计的文档化.在软件开发支持域有 1 个可信原则.

设计工具.在设计中应采用 CASE 工具以维护设计/需求的跟踪映射关系并生成设计文档.代表证据如“设计自动化环境”,是指“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的程度等级:(1) 没有自动化工具的非正式设计;(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;(5) 具有设计到代码自动对照的正式设计”,等级类型的可信证据,可信 2 级需要满足程度等级 2 级的要求,可信级别越高则需要的程度等级也逐级提高.

4) 代码开发:该阶段的主要目标是在满足软件需求和设计的条件下,保证代码的完整性、一致性和正确性,实现代码的可跟踪性和代码的文档化.在整个生命周期中,代码开发阶段基于软件源代码标准的工具和方法,分析并确定软件源代码是否满足要求.在代码开发阶段的软件开发支持域有两个可信原则,分别是:

● 源代码标准.一个明确定义的源码标准应该在编码活动中被使用.代表证据如“是否有源码标准”,即“是否定义了源代码的编程语言、方法、规则和工具”,布尔型可信证据,可信 2 级及 2 级以上为 1.

● 源代码分析工具.应该使用度量复杂度和风格的工具和步骤来分析所有开发的代码.代表证据如“是否有源代码分析工具”,布尔型可信证据,可信 2 级及 2 级以上为 1.

5) 测试:测试阶段的目标是在满足软件需求和设计的条件下,保证系统测试的完整性、一致性和正确性.在软件开发支持域有 1 个可信原则.

测试工具.软件工程环境应该包含一个创造、执行、文档化和分析测试完整性的测试工具集.代表证据如“测试工具类别的完整性”,是指“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),将其完整级别分为 3 级:(1) 低(包括类别数<3);(2) 中(包括类别数≥3);(3) 高(包括类别数≥4)”,等级类型的可信证据,可信 2 级、3 级需要满足完整等级 2 级的要求,可信 4 级、5 级需要满足完整等级 3 级的要求.

3.2 文档化

文档可信过程域是依据软件系统的可信要求,要求一些关键的开发活动和制品必须开发合适的文档,以支持软件系统开发和维护的可信性.文档化可信过程域共有 4 个可信原则,涉及需求分析与管理、软件设计、代码开发、测试 4 个开发阶段.

(1) 需求分析与管理

需求分析文档化.除了软件需求规约说明书和接口规约说明书,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的信息都应该被文档化.代表证据如“系统需求文档化”,根据系统需求内容的完整性,将这一可信证据设置为 3 个等级:(1) Low(不包含任何相关信息,或对功能性描述不够详细和准确);(2) Normal(包含功能性需求和非功能性需求的描述,但是对非功能性需求没有更具体的描述);(3) High(详细包含上面所描述的内容)”,等级类型的可信证据,可信 2 级需要满足完整性等级 2 级的要求,可信 3 级及 3 级以上需要满足完整性等级 3 级的要求.

(2) 软件设计

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该被文档化.代表证据如“过程设计文档化”,该可信证据描述为“根据是否完整、准确、具体地描述系统的过程设计,将过程设计文档化可信证据设置为 3 个等级:(1) Low(无法满足或只部分满足软件项目活动的需要);(2) Normal(刚好满足软件项目活动正常进行的需要);(3) High(设计信息记录全面,能够充分满足软件项目活动的需要)”,等级类型的可信证据,可信 2 级需要满足程度等级 2 级的要求,可信 3 级及 3 级以上需要满足程度等级 3 级的要求.

(3) 代码开发

源代码文档化.源码和软件编码活动的特征应该被文档化.代表证据如“程序内部文档化”,该可信证据描述为“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,设置为 3 个等级:(1) Low(程序内部文档标示符不恰当、缺少应有的注释以及组织混乱);(2) Normal(程序内部文档化使得源程序代码恰好可读);(3) High(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)”,等级类型的可信证据,可信 2 级需要满足文档化程度等级 2 级的要求,可信 3 级及 3 级以上需要满足文档化程度等级 3 级的要求.

(4) 测试

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该被文档化.代表证据如“集成测试文档化”,该可信证据描述为“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,将集成测试文档化可信证据设置为 3 个等级:(1) Low(无法满足或只部分满足软件项目活动的需要);(2) Normal(刚好满足软件项目活动正常进行的需要);(3) High(记录各种详细信息,能够充分满足软件项目活动的需要)”,等级类型的可信证据,可信 2 级需要满足程度等级 2 级的要求,可信 3 级及 3 级以上需要满足程度等级 3 级的要求.

3.3 可验证

验证与确认可信过程域的目的是对软件过程的阶段产品进行必要的评审验证,以确定阶段产品是否满足设计要求,了解并保障过程制品的质量.验证与确认域共有 8 个可信原则,涉及软件开发环境建设、需求分析与管理、软件设计、代码开发、测试 5 个阶段.

(1) 软件开发环境

形式化验证要求.所有的形式化规约和验证活动都应该遵循一种合适的方法,包括使用形式化规约和验证工具、文档、同行评审和可跟踪等.代表证据如“形式化可跟踪性”,即“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”,是布尔型可信证据,可信 2 级及 2 级以上为 1.

(2) 需求分析与管理:共有两个验证与确认方面的可信原则,分别是:

● 需求分析评审.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.代表证据如“软件需求正式评审与审查”,旨在判断受审制品的完整性、正确性、一致性和开发方针及标准的符合程度,证据为:(1) 一般未进行评审和审查;(2) 评审和审查是非正式的,一般效果不明显;(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”,等级类型的可信证据,可信 2 级需要满足程度等级 3 级的要求,可信级别越高则需要的程度等级也逐级提高.

● 形式化需求验证.除了非形式化的需求规约说明以外,需求文档应用一个形式化的框架来规约.代表证据如“是否有形式化需求规约”,布尔型可信证据,可信 4 级及 4 级以上为 1.

(3) 软件设计:共有两个验证与确认方面的可信原则,分别是:

● 设计评审.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.代表证据如“设计阶段评审组的经验”,即“指项目组成员在执行正式设计评审方面的平均经验.证据为:(1) Very Low(平均少于 4 个月);(2) Low(平均 4 个月);(3) Normal(平均 1 年);(4) High(平均 2 年);(5) Very High(平均 3 年);(6) Extra High(平均 4 年)”,等级类型的可信证据,可信 2 级需要满足经验等级 3 级的要求,可信 3 级及 3 级以上需要满足经验等级 4 级的要求.

● 形式化设计验证.形式化验证形式化的设计规约说明书是否满足它的需求.代表证据如“形式化设计验证范围”,是指“设计验证范围分为 3 个等级:(1) 低(无形式化验证);(2) 中(安全关键相关部分);(3) 高(全系统)”,等级类型的可信证据,可信 2 级、3 级需要满足程度等级 1 级的要求,可信级别越高则需要的程度等级也逐级提高.

(4) 代码开发:共有两个验证与确认方面的可信原则,分别是:

● 源代码评审.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.代表证据如“源代码审计策略”,即“对源代码阶段的过程产品是否有审计制度”,布尔型可信证据,可信 2 级及 2 级以上为 1.

● 形式化代码验证.形式化验证以证明底层的源码形式化规约是否满足它的需求,所有的形式化规约和验证活动都应该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.代表证据如“是否有形式化代码验证”,即“是否在代码级别使用的形式化验证技术”,布尔型可信证据,可信 5 级为 1.

(5) 测试

测试评审.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.代表证据如“对测试用例的评审”,即“测试用例是否完整”,布尔型可信证据,可信 2 级及 2 级以上为 1.

3.4 可管理

管理可信过程域共有 8 个可信原则,涉及到软件开发环境和全生命周期.

(1) 软件开发环境:软件开发环境属于预备软件开发阶段,主要目标是建立合适的软件开发环境,以确保在可信的环境下开发可信的产品.这里共有 3 个管理方面的可信原则,分别是:

● 管理政策.根据管理文档,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.代表证据如“是否有管理计划”,布尔型可信证据,可信 2 级及 2 级以上为 1.

● 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该与至少两个人员相关,这些人员应该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.代表证据如“是否有知识共享”,布尔型可信证据,可信 2 级及 2 级以上为 1.

● 环境完整性.对于识别软件工程环境组件的变更应该有一个明确的步骤,如果有需要,恢复环境的完整性.代表证据如“软件工程环境变更记录”,即“是否有软件工程环境组件变更识别的功能”,布尔型可信证据,可信3级及3级以上为1.

(2) 全生命周期:这里共有5个管理方面的可信原则,分别是:

● 计划.对于所有软件开发活动的详细计划应该在软件开发计划书中加以描述,软件开发的管理也应该遵循计划书中所描述的方法.代表证据如“是否编制了软件开发计划书”,证据类型是布尔型,可信等级 2级及以上为1.

● 风险管理.与软件开发活动相关的潜在风险都应该被明确地识别,风险移除策略应该被文档化.代表证据如“是否有风险管理计划”,布尔型可信证据,可信2级及2级以上为1.

● 度量与分析.对软件过程和制品进行了合适的度量和分析,以准确地理解过程和制品的状态,及时了解过程的偏差,以采取合理的纠正措施.代表证据如“是否对软件项目进度进行了度量”,布尔型可信证据,可信2级及2级以上为1.

● 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.代表证据如“是否对配置进行了检查和评审”,布尔型可信证据,可信2级及2级以上为1.

● 过程审计.确定的软件生命周期活动的记录应该由软件工程环境自动地登记和储存在受保护的存放处.代表证据如“能否提供可追溯性的审计数据和文件”,布尔型可信证据,可信2级及2级以上为1.

3.5 可追溯

可追溯可信过程域的目的是建立制品间的跟踪关系,以支持需求的验证、确认和产品的演化和维护.可追溯域共有4个可信原则,涉及需求分析与管理、软件设计、代码开发以及测试这4个开发阶段.

(1) 需求分析与管理

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,代表证据如“软件制品到需求的可跟踪性等级”,该可信证据描述为“软件制品到需求的可跟踪性等级:(1) 少量软件需求在概要设计、详细设计、功能模块、代码等层次,建立过可用的跟踪关系,且效果较差;(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;(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”,等级类型的可信证据,可信2级、3级需要满足2级的可跟踪性等级,可信4级、5级满足3级的可跟踪性等级.

(2) 软件设计

设计可跟踪性.设计的各方面和需求应该是互相可跟踪的.代表证据如“设计到需求的追踪程度”,属于布尔型的可信证据,可信2级及2级以上为1.

(3) 代码开发

源代码可跟踪性.所有的源码应该对于设计和计算机软件单元测试是可跟踪的,设计对于源码同样如此.代表证据如“需求到源代码的可追踪程度”,即“一个需求可以追踪到至少一块代码”,布尔型可信证据,可信2级及2级以上为1.

(4) 测试

测试可跟踪性.所用软件组件和配置项的测试对于需求是可跟踪的,源码和需求对于组件和配置项的测试同样如此.代表证据如“软件组件和配置项的测试到需求的可追踪性”,属于布尔型的可信证据,可信2级及2级以上为1.

3.6 实体安全

实体安全可信过程域的目标是保证软件开发主体、权限以及环境、信息通道等实体的可信控制,分布在软件开发全生命周期即软件开发的整个阶段中,通过建立合适的机制,保证软件开发过程的实体遵守可信的安全

和管理规范,亦即在可信的环境下,由可信的人员,操作可信的资源开发软件产品.实体安全可信过程域共有 4 个可信原则,全部贯穿软件开发全生命周期,它们分别是:

● 安全政策:所有的软件开发者执行开发活动应该遵守明确定义和增强的安全政策.代表证据如“人员审查”,即“是否对人员的安全背景进行过调查”,布尔型可信证据,可信 2 级及 2 级以上为 1.

● 安全管理策略:生命周期活动的执行需要有至少两个或两个以上有资格的开发人员的认同和参与.代表证据如“是否有共享监控”,即“对于知识共享的方式、质量和知识流动是否有监控机制”,布尔型可信证据,可信 2 级及 2 级以上为 1.

● 安全环境:软件工程环境应该包括一个明确的机制来保证生命周期的活动不会被未授权的方法所截获.代表证据如“网络服务安全性”,该可信证据描述为“常见的网络服务(如远程登录、文件传输、网页浏览)有无可信信道来提供加密、认证或完整性等保护,分为 3 等:(1) 低(几乎无保护);(2) 中(部分保护);(3) 高(全部受保护)”,等级类型的可信证据,可信 2 级需要满足安全性等级 2 级的要求,可信 3 级及 3 级以上需要满足安全性等级 3 级的要求.

● 信息安全工具支持:根据清晰定义的安全策略,所有确定的软件生活周期活动应该被软件工程环境自动控制.代表证据如“自动化工具对权限控制的支持程度”,该可信证据描述为“按照权限控制工具的自动化程度分为 3 级:(1) 没有自动化工具支持;(2) 工具半自动化;(3) 工具完全自动化是否有软件工程环境组件变更识别的功能”,等级类型的可信证据,可信 2 级需要满足自动化等级 1 级的要求,可信 3 级需要满足自动化等级 2 级的要求,可信 4 级及 4 级以上需要满足自动化等级 3 级的要求.

4 可信证据

可信原则由一组可信证据支持,这些证据分为不同的级别,表达对活动不同的可信行为要求.表 1 表示了可信原则(TP)对应的可信证据结构,其中证据等级表示该证据从该级开始要求,其满足程度可以不断提高,最高等级大于等于起始级别.对于可信等级 1 级,表示执行了有关活动,但执行过程是在无定义和管理的状态下进行的,对活动的可信没有要求,因此本模型没有 1 级可信证据.

Table 1 Evidence architecture of trustworthiness principle

表 1 可信原则的证据结构

可信原则(*TP*)	2 级证据	证据 2.1、证据 2.2…
	3 级证据	证据 3.1、证据 3.2…
	4 级证据	证据 4.1、证据 4.2…
	5 级证据	证据 5.1、证据 5.2…

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度量,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 3 级时,只需要评估各原则 3 级及 3 级以下的证据.此外,模型不要求所有的级别都有证据支持.表 2 是可信原则“形式化设计验证”的例子,但当可信要求为 3 级时,该原则下 2 级所有的证据到达可信 3 级或其本身的最高级别即可,具体在第 5 节给出介绍.

Table 2 Trustworthiness principle example of “Verification for Formal Design”

表 2 可信原则“形式化设计验证”示例

形式化设计验证	2 级证据	2.1 低层设计的形式化程度 2.2 高层设计的形式化程度 2.3 功能设计的形式化程度 2.4 形式化设计验证范围 2.5 人员对形式化设计验证工具熟练程度 2.6 形式化设计验证工具自动化程度
	3 级证据	–
	4 级证据	4.1 是否有形式化设计规约 4.2 是否有形式化设计验证
	5 级证据	–

证据(evident)是支撑可信原则的基础.证据是软件开发活动所留下的数据,经过度量分析,表明活动的表现是否满足要求.这些证据是过程可信的基本元素,也称为可信证据.

在模型中,可信证据由相关的度量支撑,如下表示.

定义 3. *Evident*=⟨*Metric*,*T-Level*,*Evident-Type*⟩.

基于证据的来源和属性,本模型将证据分为两类,一类是过程可信证据,主要度量过程的行为是否可信,支持前 36 个可信原则;另一类是制品可信证据,主要度量过程的结果,即制品是否可信,支持第 37 条可信原则.

Evident-Type∈{*Process-Evident*,*Artifact-Evident*},证据类型不同,其度量 *Metric* 的定义不同,*T-Level* 表示该证据满足的可信级别,从最低 1 级到最高 5 级共分 5 个级别.

一些证据可视项目情况进行裁剪,裁剪时需要给出裁剪理由,并确定不会影响到其他证据的实现.

4.1 软件过程可信证据

过程可信证据对应于 Process-Evident 类型,其对应的证据度量可有如下定义.

定义 4. *Metric*=⟨*MName*,*Lower-Limit*,*Upper-Limit*,*Performance*⟩,*MName* 表示度量元,*Lower-Limit* 是该度量适用的最低可信级别,在表 1 中,证据所属的级别即是该证据的 *Lower-Limit*.*Upper-Limit* 是该度量可以满足的最高可信级别,*Performance* 是该度量的实际值,介于 *Lower-Limit* 和 *Upper-Limit* 之间.

本模型共定义了 182 个过程可信证据,按照开发阶段和可信过程域分为两种表示模型,分别见表 3 和表 4.

每个证据度量的结果反映了该证据可支持的可信级别,以可信证据“人员对测试工具熟练程度”为例,该证据要求:“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的掌握程度分为 5 级:(1) 低(无工具支持);(2) 较低(个人使用,半年以内,不太熟练);(3) 一般(开发小组,半年以上,一般掌握);(4) 较高(部门内,使用超过 1 年,较熟练);(5) 高(全公司范围、使用超过 3 年,熟练掌握)”.该证据支持的可信级别见表 5.

Table 3 Process evidences grade distribution according to the development phase

表 3 按开发阶段分类表示过程证据等级分布

开发阶段	TP	证据等级					
		1 级	2 级	3 级	4 级	5 级	总计
环境	7	0	25	3	0	0	28
需求	5	0	24	1	1	0	26
设计	5	0	25	0	2	0	27
源代码	6	0	23	0	0	2	25
测试	4	0	21	7	0	0	28
全生命周期	9	0	48	0	0	0	48
合计	36	0	166	11	3	2	182

Table 4 Process evidences grade distribution according to the process area

表 4 按可信关键域分类表示过程证据等级分布

可信关键域	TP	证据等级					
		1 级	2 级	3 级	4 级	5 级	总计
实体安全	4	0	18	0	0	0	18
开发支持	8	0	34	9	0	0	43
可管理	8	0	39	2	0	0	41
文档化	4	0	29	0	0	0	29
可验证	8	0	33	0	3	1	37
可追溯	4	0	13	0	0	1	14
合计	36	0	166	11	3	2	182

例如,如果要到可信 2 级,该证据对应的掌握程度应该到 3,亦即一般掌握的水平;若希望到可信 5 级,则需要全公司范围内熟练掌握.

Table 5 Five levels of trustworthiness requirements of “People’s Proficiency in Testing Tools”

表 5 可信证据“人员对测试工具熟练程度”的 5 级可信要求

阶段名称	可信过程域名称	可信原则	对应可信证据	证据类型	1 级	2 级	3 级	4 级	5 级
测试	开发支持	测试工具	人员对测试工具熟练程度	等级	0	3	4	4	5

4.2 软件制品可信证据

对于第 37 条可信原则——制品可信,我们建立了 108 类证据,这些可信证据覆盖软件的全生命周期的制品.主要源自学术界和产业界目前已经较为广泛采用的度量.

制品可信原则,属于全生命周期阶段,并且不归属到某个可信过程域.其证据度量可定义如下.

定义 5. *Metric*=⟨*MName*,*Attribute*,*Applied-Level*,*Performance*⟩,*MName* 表示度量元,*Attribute* 是该证据对应的过程阶段,包括需求、设计、源代码、测试和产品共 5 个阶段,*Applied-Level* 是该度量适用的可信级

别,*Performance* 是该度量实际值.

由于软件过程制品的度量数量繁多,很多还具备可替代性,譬如缺陷率、缺陷密度等等,模型并不强调一定要用哪个,只是给出证据的基本要求和可能的选择.已经建立的 108 个制品证据主要来自 ISO 9126 以及业界普遍使用的证据度量.同时,我们将 108 个制品可信证据分为 46 类,对应到需求、设计、源代码、测试、产品这 5 个阶段,同一类同一级下的可信证据可以选择 1 个即可.表 6 是制品可信证据按阶段的分类明细.

Table 6 Classification table according to measurement phase of artifacts evidences

表 6 按照制品可信证据度量的阶段分类表

阶段	证据类数	证据场景等级					
		1 级	2 级	3 级	4 级	5 级	总数
需求	5	2	0	2	3	0	7
设计	3	1	1	1	3	0	6
源代码	5	1	0	4	4	1	10
测试	6	1	1	5	7	2	16
产品	27	30	0	18	15	6	69
合计	46	35	2	30	32	9	108

表 6 中的证据场景是指该证据在所属的场景的等级需要.在证据场景等级的界定方面,我们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 1 级(原型与实验级)到最高风险等级 5 级(生命攸关级)逐级划分(见表 7).例如,如果某可信证据关系到灾难性损失,就会界定为 5 级的证据.表 7 的其他分类将在后续作可信评估时用到.

Table 7 Grading table of product failure severity

表 7 产品失效严重性分级表

	1 级	2 级	3 级	4 级	5 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 OA	商业办公软件	金融系统(支付宝、网银系统)	航空航天高铁

5 软件过程可信评估

本可信度模型建立了软件过程可信的基本要求和支撑证据.基于本模型,在软件开发过程中采集相应的数据,并进行度量,即可获得需要的证据,并进一步对软件过程的行为和制品的可信性进行评估.

5.1 证据表示的可信级别

如定义 3 所示,*Evident.T-Level* 表示该证据满足的可信级别.如前所述,本模型的证据分为两类,当 *Evident.Evident-Type=Process-Evident* 时,表示该证据用于评价软件过程行为的可信度,其可信证据的度量见定义 4.对此类任意证据 *Evident*,*T-Level* 的判定方法如下:

$$T\text{-}Level=\begin{cases}1, & \text{if } Metric.Performance < Metric.Lower\text{-}Limit\\ Metric.Performance, & \text{if } Metric.Lower\text{-}Limit \le Metric.Performance < Metric.Upper\text{-}Limit\\ Metric.Upper\text{-}Limit, & \text{if } Metric.Performance \ge Metric.Upper\text{-}Limit\end{cases}.$$

当 *Evident.Evident-Type=Artifact-Evident* 时,表示该证据用于评价过程制品的可信度,其可信证据的度量见定义 5.对此类任意证据 *Evident*:

$$T\text{-}Level=\begin{cases}1, & \text{if } Metric.Performance < Metric.Applied\text{-}Level\\ Metric.Applied\text{-}Level, & \text{if } Metric.performance \ge Metric.Applied\text{-}Level\end{cases}.$$

根据证据的实际度量数据得到的证据,表示该证据表示的过程行为满足的可信程度,是过程可信评估的基础.这些基础的证据可以实现量化评估软件的可信性,避免评估人员主观定性的判断带来的可信性度量和评估结果不准确、不可靠等问题.

5.2 可信原则的可信度评估

在本模型中,可信原则的表示见定义 2.可信原则满足的可信程度取决于其下属的可信证据所表征的可信级别.

我们设计一种算法来计算单个可信原则的可信等级 *S-Level*,引入可信原则列表和证据列表,*TP.ESet* 表示该可信原则 *TP* 的可信证据集合.这里,我们遵循木桶原理,亦即用该集合中最低的证据级别来表示该原则的级别.此外,我们还需要考虑该原则下每个可信证据等级 *Evident.T-Level* 和 *Evident.Metric.Upper-Limit* 的大小关系.可信原则达到的可信级别的评估方法原则如下.

算法 1. 可信原则的可信等级评价方法.

INPUT:

TP.ESet; //可信原则 *TP* 的可信证据列表

OUTPUT:

S-Level; //可信原则的可信等级

BODY:

1: *S-Level*=1

2: **IF** 对于所有 **2** 级证据

3: $Evident_i.T.Level \geq 2$

4: **THEN** *S-Level*=2

5: **IF** 对于所有 **2+3** 级证据

6: $Evident_i.T.Level \geq 3$ or $Evident_i.T\text{-}Level=Evident.Metric.Upper\text{-}Limit$

7: **THEN** *S-Level*=3

8: //到 **3** 级或 **2** 级证据为最高等级

9: **IF** 对于所有 **2+3+4** 级证据

10: $Evident_i.T.Level \geq 4$ or $Evident_i.T\text{-}Level=Evident.Metric.Upper\text{-}Limit$

11: **THEN** *S-Level*=4

12: //到 **4** 级或 **2**、**3** 级证据为最高等级

13: **IF** 对于所有 **2+3+4+5** 级证据

14: $Evident_i.T.Level \geq 5$ or $Evident_i.T\text{-}Level=Evident.Metric.Upper\text{-}Limit$

15: **THEN** *S-Level*=5

16: //到 **5** 级或 **2**、**3**、**4** 级证据为最高等级

下面我们以表 8 所示的可信原则 TP_{25} 为例.

Table 8 Rating results of eight trustworthiness evidences of TP_{25}

表 8 TP_{25} 所属的 8 个可信证据的等级评估结果

TP	证据级别	证据	*Lower-Limit*	*Upper-Limit*	*Performance*	*T-Level*	*S-Level*
TP_{25}: 形式化设计验证	2 级	2.1 低层设计的形式化程度	2	4	4	4	2
		2.2 高层设计的形式化程度	2	5	5	5	
		2.3 功能设计的形式化程度	2	5	5	5	
		2.4 形式化设计验证范围	2	5	2	2	
		2.5 人员对形式化设计验证工具熟练程度	2	3	2	2	
		2.6 形式化设计验证工具自动化程度	2	5	2	2	
	3 级	–	–	–	–	–	
	4 级	4.1 是否有形式化设计规约	4	4	1	1	
		4.2 是否有形式化设计验证	4	4	1	1	
	5 级	–	–	–	–	–	

需要注意的是,当评估的可信需求不一样时,可信原则对应的可信证据集合也不一样.在表 8 所示的例子中,如果对某产品或过程的可信等级要求是 3 级或 3 级以下,则 $TP_{25}.ESet=\{E_{2.1},E_{2.2},E_{2.3},E_{2.4},E_{2.5},E_{2.6}\}$;当可信等级要求是 4 级或 4 级以上时,$TP_{25}.ESet=\{E_{2.1},E_{2.2},E_{2.3},E_{2.4},E_{2.5},E_{2.6},E_{4.1},E_{4.2}\}$.根据算法 1 的评估结果,$TP_{25}$ 的可信等级 *S-Level* 为 2.

5.3 过程的可信度评估

基于对可信原则满足程度的评估,我们可以进一步评估整个软件过程实现的可信度水平.对软件过程的可信度评估遵循木桶原理,亦即在 *Required-Level* 指定的可信范围内,所有适用的原则必须达到要求的可信级别,亦即

当且仅当

对所有可信原则 *TP*,若 *TP.S-Level*≥*SPTModel.Required-Level*

or

该 *TP* 判定为不适用,则:

SPTModel.Satisfied=True

否则

SPTModel.Satisfied=False

对于在模型应用时判定为不适用的可信原则,评估时不作评价.可信原则的裁剪准则,我们不在本文范围内讨论.

6 应用案例

根据以上介绍的模型和方法,软件组织或项目可以按以下步骤建立可信的软件过程模型,并依据采集的证据数据,对过程的可信度进行评估.

第 1 步:根据软件产品的应用领域和目标,建立软件开发要求的过程可信级别.

第 2 步:基于模型的证据要求和项目实际情况,进行裁剪,建立组织或项目适用的证据集合.

第 3 步:根据证据要求,收集项目数据.

第 4 步:根据采集到的数据,基于证据的度量方法,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.

第 5 步:依据可信原则的评估方法,评估可信原则实现的可信等级.

第 6 步:依据可信模型的评估方法,评价软件过程是否满足预期的可信要求.

为了支持项目组织进行有效的证据分析和评估,我们开发了一个评估支持工具,实现可视化地给出软件组织或者项目的可信证据视图.软件组织和项目可以利用该工具,方便地进行证据分析和过程评估,并可进行差距分析和识别改进的区域.

证据视图通过不同颜色的色块分布,展示项目数据对应的过程可信证据的等级情况.

图 3 所示为过程可信证据视图采用的图例,不同颜色的方块表示该证据的不同可信等级,其中红色方块表示该可信证据未满足可信要求(即到可信一级);灰色表示该证据在实验项目中不适用,未参与可信等级评估;而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.

在过程证据视图中,横坐标表示按照开发阶段或者可信过程关键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.

对应不同的可信要求,如果其级别及其下面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 3 左侧证据视图中没有红色色块,所有过程证据可信等级满足最小为 2 级的可信要求,整个项目的过程可信等级为 2.当 2 级和 3 级可信证据分别满足该证据的最高等级,且其他证据的取值提高到满足可信 4 级要求时,该项目的过程可信等级为 4 级,如图 3 右侧证据视图所示.

图 4 所示为制品可信证据视图采用的图例,目前制品证据指标只有“达标”和“不达标”两种取值,其中红色

方块表示该可信证据未达到可信要求,而灰色表示该可信证据在实验项目中不适用.

在制品证据视图中,横坐标表示按照阶段划分的所有证据指标类,纵坐标表示每个等级下的可信证据个数.制品的可信等级表示单个可信指标从该等级开始要求可信(不论是否达标).

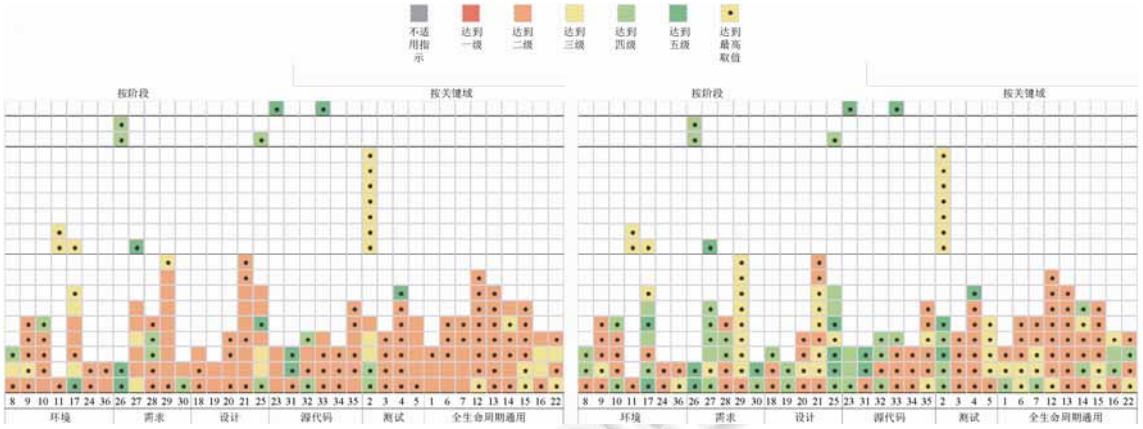

Fig.3 Comparison in evidences view between level 2 and level 4 of process trustworthiness

图 3 项目过程可信 2 级和可信 4 级的证据视图对比

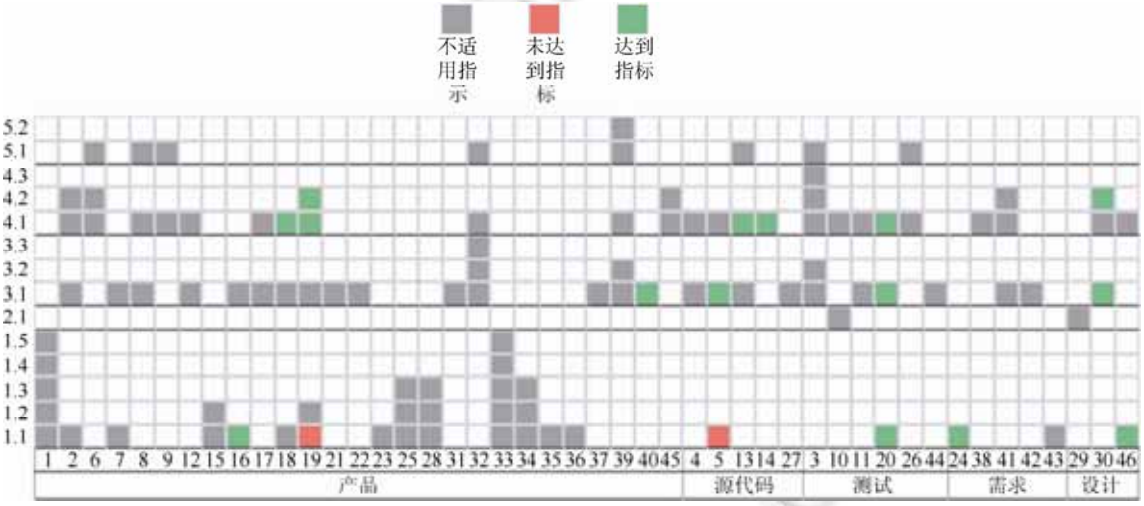

Fig.4 View of artifacts evidences

图 4 制品可信证据视图

在过程可信视图和制品可信视图中,点击任何方块均可弹出该证据指标的等级说明.

6.1 案例背景

本节将以两个实际的软件开发项目作为案例进行分析和验证.案例来自高校的工程项目和我们在产业合作中实际应用的工业项目,两个案例都是嵌入式软件系统领域.

下面我们详细展示两个案例项目的证据视图,包括过程视图和制品视图,根据不同项目的可信等级要求,对统计结果进行分析并提出改进的建议.

6.1.1 项目 A

该项目面向航空保障信息系统的嵌入式系统和构件开发,采用 CMMI ML5 级标准和瀑布式软件生命周期模型,应用单位通过了 GJB 9001 认证,项目周期为 5 年,该项目的可信等级要求为 5 级.

图 5 是该项目的过程证据视图和制品证据视图.

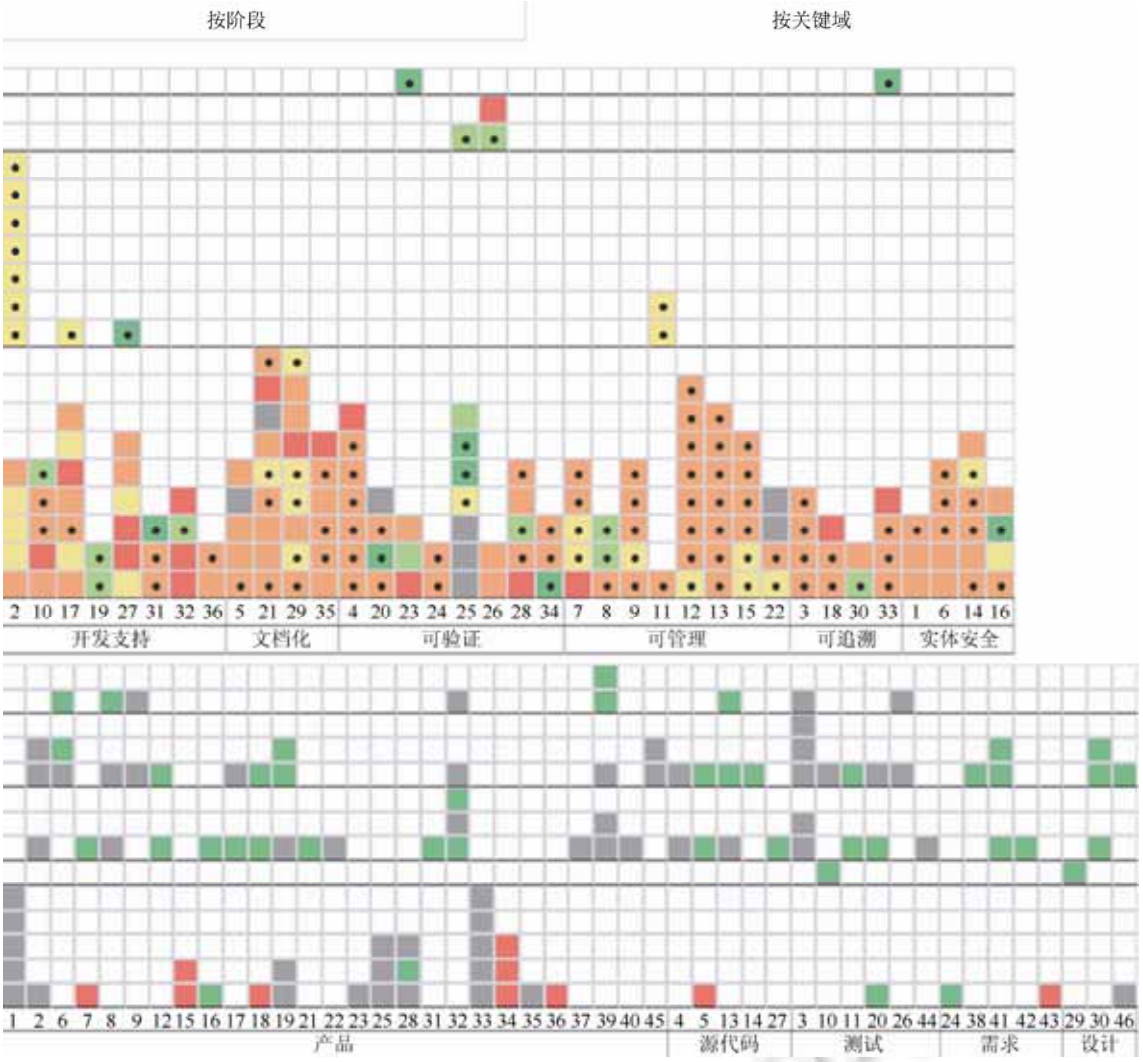

Fig.5 View of process evidences and artifacts evidences in Project A

图 5 项目 A 的过程可信证据视图和制品可信证据视图

该项目的证据视图表示过程为可信 1 级.可信证据等级结果分析和改进建议如下.

(1) 约 58%的 2 级过程证据达到最高等级或者满足可信 5 级,3 级及 3 级以上证据表现良好,仅有 1 个证据未满足可信 5 级要求,说明其大部分的过程证据满足项目 5 级的可信要求.

(2) 部分 2 级证据未达标,拖累了整个过程的可信水平.2 级证据未达标的可信原则主要集中在开发支持和文档化两个可信过程域.在开发支持域,需求分析和源代码两个阶段的支持工具不足;在文档化过程域,每个阶段都有少量的证据没有达到 2 级要求,需要在文档的有效性和合理性方面加以完善,而不是仅仅限于有文档即可.另外,在可验证、可管理和可追溯过程域,都存在少数证据等级低的情况,表现在人员对形式化代码验证工具熟练程度、自动化工具对设计可跟踪性和源代码可跟踪性的支持程度等方面.结合开发支持和文档化两个可信过程域的证据表现,说明在该项目中,自动化工具支持和人员对相关工具的熟练程度是两大薄弱方面,需要重点加强.

(3) 整个项目不适用的过程证据较少,只占到总数的 4%,说明我们建立的过程可信度模型和可信证据体系能够大体反映实际项目所要求的可信属性和可信等级.

(4) 在制品可信证据方面,所有的 2 级和 2 级以上的证据或者满足或者不适用,而 1 级证据表现不好,说明一些基础工作没有做好.当然,另一方面也提示我们在制品证据的选择和建模方面应该进一步加以改进和优化.

根据项目 A 应用单位的反馈,我们的可信等级分析结果符合项目实际,能够体现项目过程和制品可信的薄弱环节,说明我们提出的可信度模型和评估方法具有一定的实用性.

6.1.2 项目 B

该项目面向专用嵌入式系统软件,采用 CMMI ML3 级标准和瀑布式软件生命周期模型,应用单位通过了 GJB 5000A(CMMI)3 级认证,项目周期为 4 年,该项目的可信等级要求为 3 级.

图 6 所示为该项目的过程证据视图和制品证据视图.

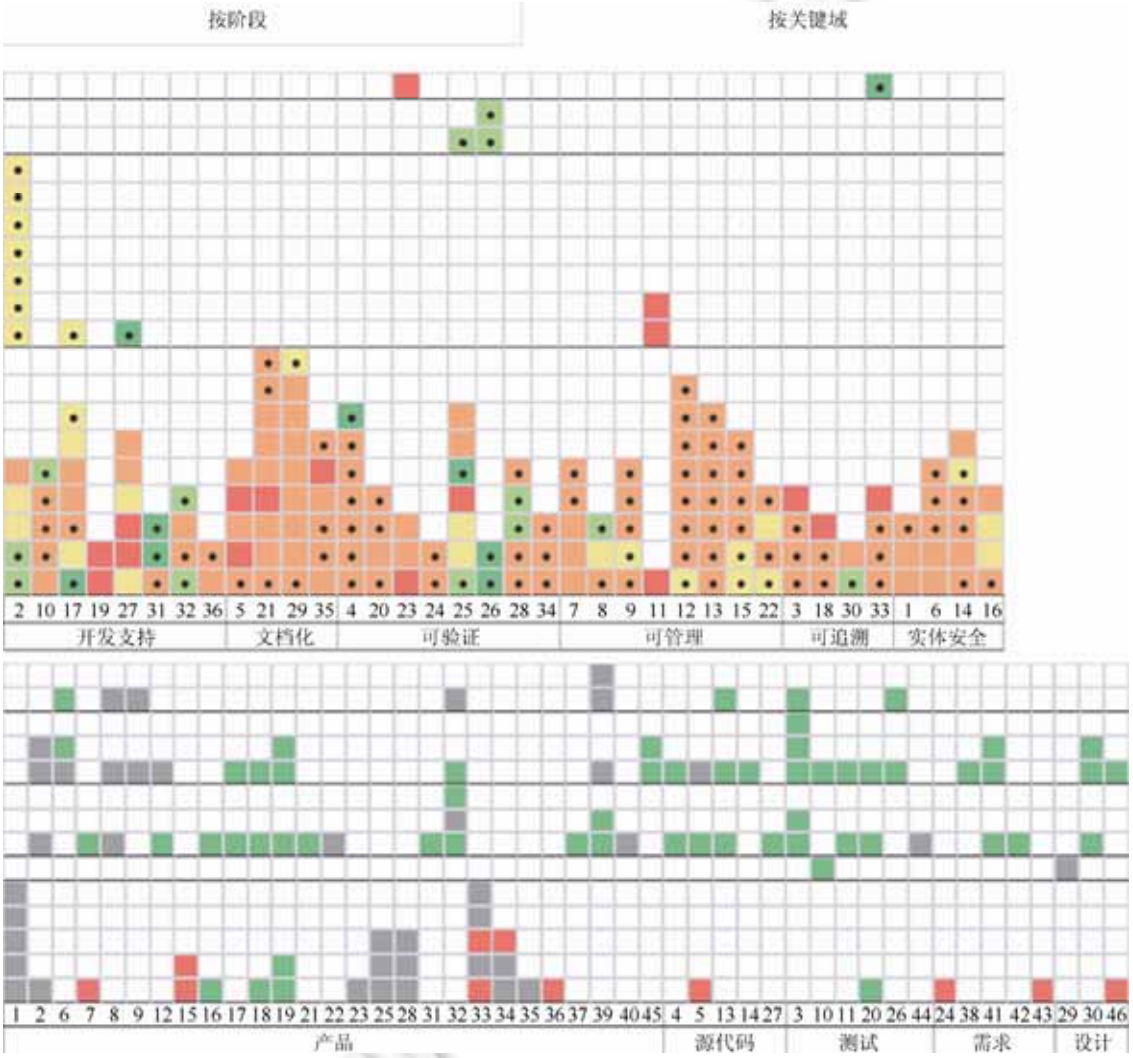

Fig.6 View of process evidences and artifacts evidences in Project B

图 6 项目 B 的过程可信证据视图和制品可信证据视图

该项目的证据视图表示过程为可信 1 级,也没有达到目标等级.可信证据等级结果分析和改进建议如下.

(1) 约 70%的 2 级过程证据达到最高等级或者满足可信 3 级,而 3 级及 3 级以上证据中仅有 3 个证据未达到可信 2 级,证明其大部分的过程证据满足项目 3 级的可信要求.

(2) 部分 2 级证据未达标,拖累了整个过程的可信水平.2 级证据未达标的可信原则主要集中在文档化和可追溯两个可信过程域.在文档化过程域,调试文档、集成测试文档、数据库设计结果这些文档的水平有待改进;在可追溯域,自动化工具对测试可跟踪性的支持程度、自动化工具对设计可跟踪性的支持程度、自动化工具对源代码可跟踪性的支持程度不够.结合开发支持过程域未满足的证据,说明在支持工具和开发环境建设方面存在差距.

(3) 没有不适用的过程证据,说明我们建立的过程可信度模型和可信证据体系能够反映该项目要求的所有可信属性和可信等级.

(4) 与项目 A 类似的情况,在制品可信证据方面,所有的 2 级和 2 级以上证据或者满足或者不适用,而 1 级证据表现不好.1 级证据中约 54%被判断为不适用,集中在产品阶段,主要因为该项目不采集相关数据.大部分适用的 1 级制品证据没有达标,拖累了整个制品可信的等级,这与案例 A 的情况类似.

根据项目 B 应用单位的反馈,我们的可信等级分析结果符合项目实际,能够体现项目过程和制品可信的薄弱环节.由于项目 B 没有不适用的过程证据,说明我们建立的过程可信度模型和可信证据体系能够反映该项目要求的所有可信属性和可信等级,可信度模型和评估方法具有实用性和一定的适用性.

7 总结与讨论

本文工作从软件开发过程的角度,探讨过程实体、过程行为和过程制品对软件产品可信性的影响,并基于覆盖软件全生命周期的证据,建立软件过程可信度模型,希望从软件开发的过程,系统地建立最终产品的可信证据链,为软件产品可信提供客观系统的信心.

本节将从过程可信评估方法的特点以及案例分析的有效性威胁两个方面进行论述.

7.1 模型特点

本文提出的模型包含 37 条可信原则,182 个表征过程实体和行为可信的证据,108 个表征过程制品可信的证据,每一条可信原则都由一组证据支撑.该模型是一个开放的模型,软件项目和组织可以根据具体的可信要求选择合适的证据子集.此外,本模型建立的可信原则和证据集合都可以根据实际要求裁剪和增加,模型的特点总结如下.

(1) 基于客观的过程数据.本方法基于的过程可信证据来自于实际的过程度量数据,通过证据建模得到支持评估的客观证据.因此,评估方法能够反映出软件开发过程中实际的可信状况,得到的等级分布符合项目开发的实际情况,是客观公正的.项目合作单位可以根据相应的问题和薄弱点进行过程可信改进和提高.

(2) 评估的全面性.相对于基于 CMMI 和 TSM 的可信评估方法而言,本文的评估方法在 CMMI 和 TSM 的基础上进行了可信证据的扩展,形成了较为完整的过程可信开发阶段和可信过程域,保持了可信评估方法的全面性.

7.2 与CMMI成熟度级别的关系

从两个案例看,对于一个采用 CMMI ML3~5 进行过程改进的组织,依然有部分可信原则达不到 2 级的要求,这是因为,CMMI 强调过程的可管理性,在大多数 CMMI 的评估中,只观察是否有证据以及是否有弱项,而对证据本身的要求以及证据表现的性能,并不关注.譬如在开发支持阶段,CMMI 关注足够的资源支持开发活动,但评估时不会强调工具的覆盖范围.但在可信度模型中,则对工具的支持程度有不同的等级要求,以对应不同的可信要求.这也使得很多同样成熟度级别的组织,其过程能力差距其实很大.本模型强调管理的可信性,从实体、行为、制品 3 个保障目标定义并明确了过程实践应该提供的证据,使得基于本模型的评估更加客观,也具有更好的可比性.此外,本可信度模型在建立可信度要求时,允许用户根据所开发产品的可信要求和失效风险确定过程应该达到的可信等级,而 CMMI 只是依据过程的成熟程度来确定其目标等级,与产品本身的特点无关.也就是说

CMMI 本身并不关注组织的过程应该达到什么等级,CMMI ML5 级的企业可以只开发普通的民生软件,无需达到较高的可信级别.而开发航空航天软件的企业也可以只达到低级别.

7.3 有效性威胁

本研究模型中建立的制品证据主要来自 ISO 9126、文献调研和调查问卷,其中问卷的反馈较少,我们在选择时主要考虑这些度量的成熟性和数据的易获得性,但应用的广泛性尚需进一步加以验证,证据中采用的度量可能有一定的局限性,这也是本模型目前最主要的弱点.此外,关于证据的裁剪性指南,还需要进一步制定和完善.本文介绍的研究成果正在申请国家标准,下一步我们将邀请更多软件开发组织和测试机构,参与本模型的完善和标准的制定工作,特别是其中可信证据度量支持指标的选择和确定部分,以加强模型的成熟度、实用性和适用性.

7.4 未来的研究工作展望

在本文研究成果的基础上,未来我们还将针对以下几项内容进行进一步的工作.

(1) 证据支撑度量指标的完善和优化.软件过程及其制品已经存在大量的度量,事实上没有绝对适用的度量,大多数组织都是根据自身的特点、成熟能力和过程工程师的知识水平选择偏好的度量.ISO 9126 给出的内部、外部、使用度量也并非所有企业都可以接受.下一步,我们将在更大的范围征求意见和反馈,以尽可能地建立可以广泛接受的证据度量.

(2) 开放的软件开发环境,对软件的质量属性和可信提出了新的挑战.开放/开源的软件依然有可信的要求,如何调整模型的可信原则和证据体系,使之适应开放、开源的软件开发过程,并支持对开放/开源环境下的软件可信性进行评估,将是下一步非常具有挑战性的工作.

总之,信息技术使得社会对软件的需求急剧增长,软件越来越复杂、越来越庞大,而同时人们对软件质量的要求却越来越高,对质量问题的容忍度越来越低.人们不仅希望软件好用,还希望它安全、可靠、不泄露隐私,要求的质量属性越来越多,可信赖地使用软件已成为软件社会的重要诉求.所以,系统地建立软件可信的证据,并贯彻到软件开发的生命周期,不仅可以支持软件的相关利益方建立软件的可信信心,还可以帮助开发者改进其过程,以达到可信的要求.

References:

[1] Boehm BW. A view of 20th and 21st century software engineering. In: Proc. of the 28th Int'l Conf. on Software Engineering. 2006. 12–29. [doi: 10.1145/1134285.1134288]

[2] Amoroso E, Taylor C, Watson J, Weiss J. A process-oriented methodology for assessing and improving software trustworthiness. In: Proc. of the 2nd ACM Conf. on Computer and Communications Security. Virginia, 1994. 39–50. [doi: 10.1145/191177.191188]

[3] Chen HW, Wang J, Dong W. High confidence software engineering technologies. Chinese Journal of Electronics, 2003,31(S1): 1933–1938 (in Chinese with English abstract). [doi: 10.3321/j.issn:0372-2112.2003.z1.001]

[4] CNSS. Software 2015: A National Software Strategy to Ensure U.S. Security and Competitiveness. 2005.

[5] Hasselbring W, Reussner R. Toward trustworthy software systems. Computer, 2006,39(4):91–92. [doi: 10.1109/MC.2006.142]

[6] Karen MG, Theodore W, Holly LM, Lyndon O, Michael C, Thomas M, Elaine F, Robert V.Software security assurance: A state-of-the-art-report. Technical Report, DACS and IATAC, 2007.

[7] Department of Defense, National Computer Security Center. Trusted computer system evaluation criteria. DoD 5200.28-STD. 1985.

[8] Parnas DL, Van Schouwen AJ, Kwan SP. Evaluation of safety-critical software. CACM, 1990,33(6):636–648. [doi: 10.1145/78973.78974]

[9] ISO. ISO/IEC15408-Information Technology-Security Techniques-Evaluation Criteria for IT Security. 2005.

[10] Cai SB, Zou YZ, Shao LS, Xie B, Shao WZ. Framework supporting software assets evaluation on trustworthiness. Ruan Jian Xua Bao/Journal of Software, 2010,21(2):359–372 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3786.htm [doi: 10.3724/SP.J.1001.2010.03786]

[11] Tan T, He M, Yang Y, Wang Q, Li MS. An analysis to understand software trustworthiness. In: Proc. of the the 2008 Int'l Symp. on Trusted Computing. 2008. [doi: 10.1109/ICYCS.2008.484]

[12] Zeng J, Sun HL, Liu XD, Deng T, Huai JP. Dynamic evolution mechanism for trustworthy software based on service composition. Ruan Jian Xue Bao/Journal of Software, 2010,21(2):261−276 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3735.htm [doi: 10.3724/SP.J.1001.2010.03735]

[13] Wang J, Chen YX, Gu B, Guo XY, Wang BH, Jin SY, Xu J, Zhang JY. An approach to measuring and grading software trust for spacecraft software. Scientia Sinica Techologica, 2015,45(2):221−228 (in Chinese with English abstract). [doi: 10.1360/N092014-00479]

[14] SEI, CMU. Capability Maturity Model. CMU Press, 2011.

[15] Linda I, Joe J, Matt A, Roger B, Paul C, Mary H, Larry L, Curt W. Safety and security extension for integratedcapability maturity model. United States Federal Aviation Administration, 2004.

[16] ISO/IEC. Quality management principles. 2015.

[17] ISO/IEC. Software engineering—Product quality. 2011.

附中文参考文献:

[3] 陈火旺,王戟,董威.高可信软件工程技术.电子学报,2003,31(12A):1933−1938. [doi: 10.3321/j.issn:0372-2112.2003.z1.001]

[10] 蔡斯博,邹艳珍,邵凌霜,谢冰,邵维忠.一种支持软件资源可信评估的框架.软件学报,2010,21(2):359−372. http://www.jos.org.cn/1000-9825/3786.htm [doi: 10.3724/SP.J.1001.2010.03786]

[12] 曾晋,孙海龙,刘旭东,邓婷,怀进鹏.基于服务组合的可信软件动态演化机制.软件学报,2010,21(2):261−276. http://www.jos.org.cn/1000-9825/3735.htm [doi: 10.3724/SP.J.1001.2010.03735]

[13] 王婧,陈仪香,顾斌,郭向英,王保华,金晟毅,徐建,张居阳.航天嵌入式软件可信性度量方法及应用研究.中国科学:技术科学,2015,45(2):221−228. [doi: 10.1360/N092014-00479]

王德鑫(1985－),男,山东青岛人,博士,主要研究领域为可信软件,需求协商,开源社区知识共享.

王青(1964－),女,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为软件过程方法与技术,经验软件工程.

贺劼(1976－),男,高级工程师,主要研究领域为软件工程,项目管理.