

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					总数
		1 级	2 级	3 级	4 级	5 级	
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:

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

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*:

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

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 $S\text{-Level}$,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 $Evident.T\text{-Level}$ 和 $Evident.Metric.Upper\text{-Limit}$ 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:

$S\text{-Level}$; //可信原则的可信等级

BODY:

```

1:       $S\text{-Level}=1$ 
2:      IF 对于所有 2 级证据
3:           $Evident_i.T\text{-Level} \geq 2$ 
4:          THEN  $S\text{-Level}=2$ 
5:      IF 对于所有 2+3 级证据
6:           $Evident_i.T\text{-Level} \geq 3$  or  $Evident_i.T\text{-Level}=Evident.Metric.Upper\text{-Limit}$ 
7:          THEN  $S\text{-Level}=3$ 
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:          $Evident_i.T\text{-Level} \geq 4$  or  $Evident_i.T\text{-Level}=Evident.Metric.Upper\text{-Limit}$ 
11:         THEN  $S\text{-Level}=4$ 
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:          $Evident_i.T\text{-Level} \geq 5$  or  $Evident_i.T\text{-Level}=Evident.Metric.Upper\text{-Limit}$ 
15:         THEN  $S\text{-Level}=5$ 
16:         //到 5 级或 2、3、4 级证据为最高等级
    
```

下面我们表 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 $S\text{-Level}$ 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 $Required\text{-Level}$ 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 $TP.S\text{-Level} \geq SPTModel.Required\text{-Level}$

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).

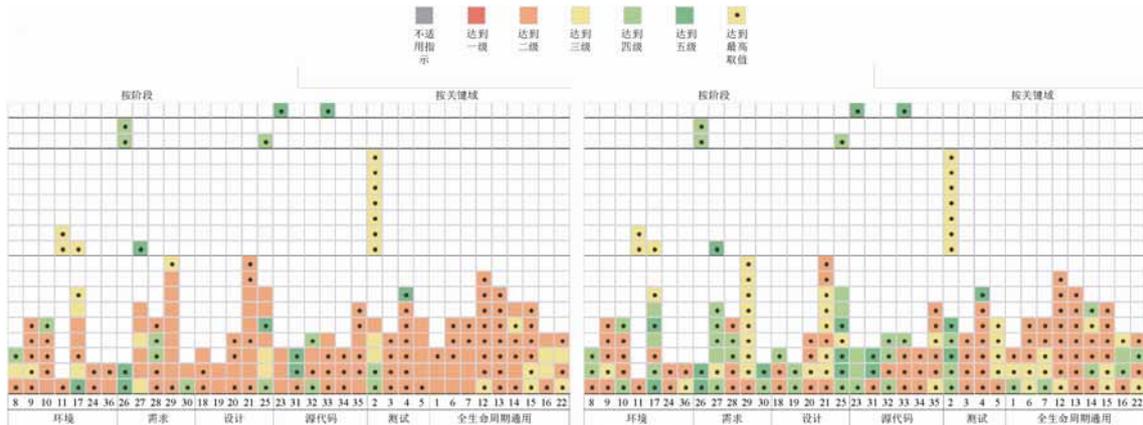


Fig.3 Comparison in evidences view between level 2 and level 4 of process trustworthiness

图 3 项目过程可信 2 级和可信 4 级的证据视图对比

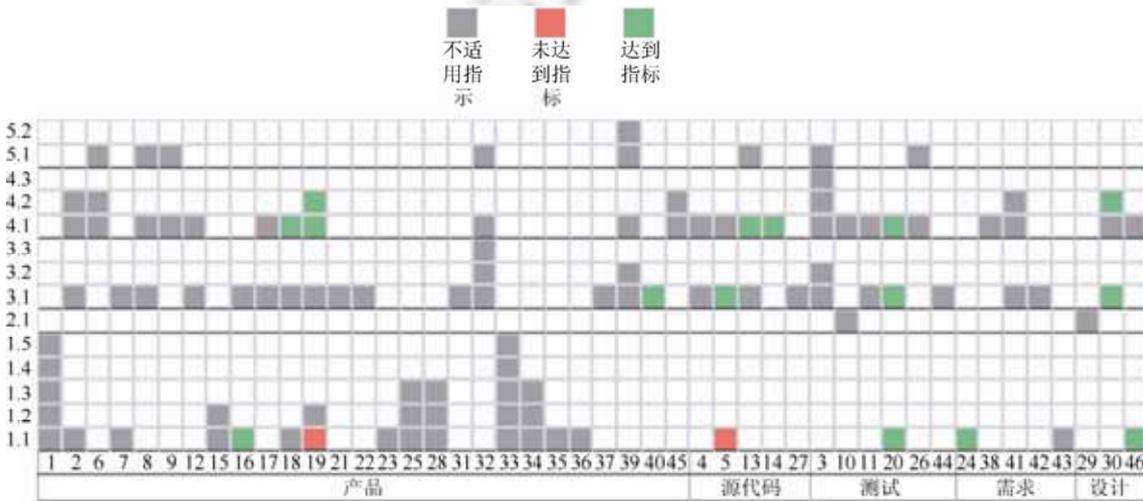


Fig.4 View of artifacts evidences

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.

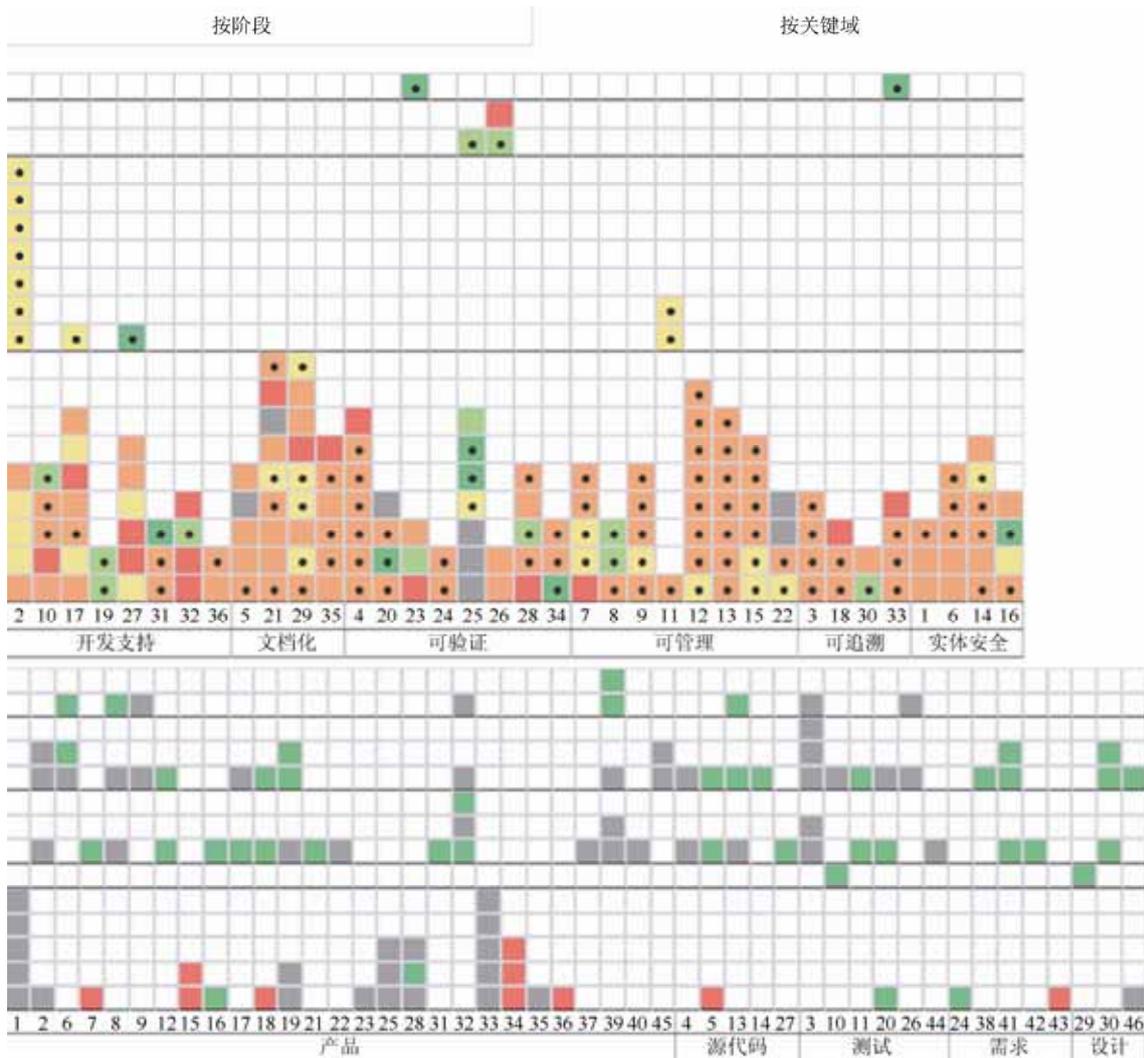


Fig.5 View of process evidences and artifacts evidences in Project A
图 5 项目 A 的过程可信证据视图和制品可信证据视图

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.

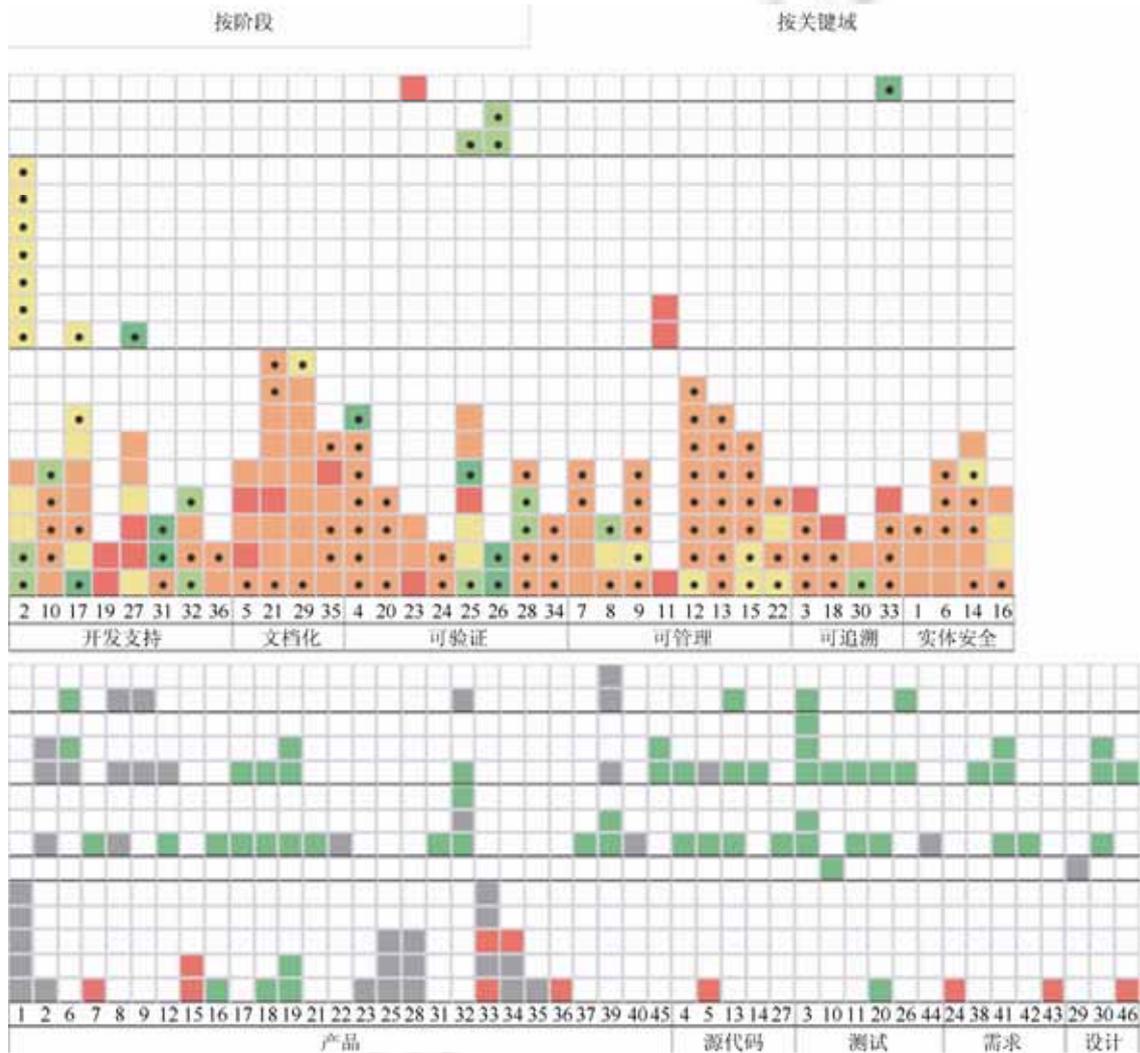


Fig.6 View of process evidences and artifacts evidences in Project B

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 Xue 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 Technologica, 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 integrated capability 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.



贺劼(1976 -),男,高级工程师,主要研究领域为软件工程,项目管理.



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.