© 2027年施普林格。本文档为预印本。允许个人使用本材料。任何其他用途均须获得施普林格许可，无论是在当前还是未来的任何媒介中，包括为广告或宣传目的重印/再版本材料、创建新的合集作品、转售或重新分发至服务器或列表，或在其他作品中重复使用本作品中任何受版权保护的组成部分。DOI：尚未提供

arXiv:2611.32809v1 [cs.SE] 24 Nov 2026

用例表如何助力获取解释需求

Lucas Burgath Jovita Vahlhaus Jakob Droste Hannah Deters
Kurt Schneider

德国汉诺威莱布尼茨大学软件工程组
lucas.burgath@inf.uni-hannover.de

摘要。 **[背景与动机]** 随着软件系统日益融入人们的日常生活——例如驾驶辅助系统或智能家居应用——它们也变得愈发复杂。使用复杂软件时，可能会产生解释需求。目前，解释需求通常通过测试原型或现有软件来获取。然而，软件交互的详细文档（例如用例表）或许是合适的替代方案。由于用例能够让人们提前了解系统内的关键交互，因此将其用作获取解释需求的资源颇具吸引力。 **[问题]** 目前尚不清楚用例能否用于获取解释需求。换个角度说，也不清楚将用例用于此目的是否可取。 **[主要观点/结果]**我们开展了一项研究，让用户阅读从现有软件中提取的用例描述部分。在阅读这些用例时，用户报告了自己遇到的任何解释需求。研究结果表明，参与者阅读用例步骤时，这些步骤足以引发解释需求。 **[贡献]** 本文探讨了利用用例获取解释需求这一新颖思路，并报告了一项研究，该研究揭示了这一概念对利益相关者的适用性。

关键词： 需求工程 · 可解释性 · 用例

1 引言

非确定性、不透明的系统组件（例如人工智能（AI）功能）的集成，使可解释性作为一项非功能需求（NFR）受到关注 [

1,7]。本质上，可解释性描述的是软件提供关于自身的解释的能力 [7,14]。当代研究已确认，解释是提升复杂系统可信度的驱动因素 [1]。通过为内部过程和系统输出提供理由，解释有助于提升软件的可理解性以及整体用户体验（UX） [10,14]。然而，早在人机交互（HCI）成为明确的研究领域之前，相关研究就已利用解释来引导交互并

提升可学习性 [21]。此外，解释也已在隐私和安全等软件工程领域得到研究，以促进透明且可信的系统设计 [4,30]。因此，解释需求可能涉及系统的各种方面，在研究与实践中仍是一个重要议题。

谨慎地实现解释并非易事。对解释的需求具有很强的主观性，某个解释是否称得上“好”，取决于接收解释的个人 [29]。与软件进行复杂交互时，往往需要多个步骤，而每个步骤都可能需要解释。利益相关者在被询问时，可能无法回忆起产生解释需求的确切时刻。为避免假设偏差和过度依赖默会知识，可通过记录相关软件交互的文档来支持解释需求的获取。然而，目前关于解释需求的研究主要集中于测试原型或现有系统，尚未考虑使用用例等更传统的文档 [6,11]。如果基于文档获取解释需求是可行的，那么这也可能使人们能够在开发周期早期考虑可解释性，此时原型尚不可用。

为弥补这一研究空白，我们对 13 名参与者开展了一项研究。我们希望考察，仅使用用例文档来获取软件系统的解释需求是否可行。我们事先为两种不同的软件应用程序准备了六个用例。一种软件是用于设计电网的专业应用程序（*Envilio*），另一种则用于全面支持组织工作（*Microsoft Teams*）。我们请参与者阅读全部十二个用例，并在描述步骤中标注解释需求。用例借助一款专用软件工具（用例工具）进行展示和标注；我们将该工具作为补充材料提供。两名研究人员检查了参与者所做的标注，识别出 153 项解释需求，且研究人员之间达成了较高的一致性。我们观察到两种应用程序之间存在两项主要差异：不仅两种系统引发的需求数量差异显著，所识别出的需求类型也各不相同。这表明，我们采用用例来获取解释需求的方法，能够揭示不同软件系统和场景中的多样化需求。

本文对可解释系统的需求工程作出了两项主要贡献：

– 一项关于利用用例获取解释需求的初步分析——可在*Zenodo* 上获取（用于在*Figshare* [2]上进行双盲评审）。– 研究中用于通过用例获取解释需求的用例工具 ——可在*GitLab* 上获取（用于在*Figshare* [2]上进行双盲评审）。

本文其余部分的结构如下：首先，我们在第 2 节介绍背景信息，并将我们的观点与现有研究加以区分。第 3 节描述我们的研究目标以及所遵循的流程。随后，我们在第 4 节报告研究结果，并在第 5 节对其进行讨论。第 6 节总结本文的研究发现，并展望未来工作。

2 背景与相关工作

2.1 可解释性与解释需求

作为一项非功能需求，可解释性描述了软件解释自身任何方面的能力 [5,14]。近年来，关于软件可解释性的研究主要聚焦于可解释人工智能（XAI） [1]。通常，这些用例中的解释用于证明输出的合理性，并实现透明且可理解的系统行为。在以人工智能为重点的应用程序之外，日常软件使用中的解释通常用于引导交互并提供可学习性 [14]。这些需求更符合传统的人机交互（HCI）研究，该研究利用解释来提高心智模型准确性和用户效率 [21]。理解这些需求并将其形式化为可执行的系统需求，是可解释系统需求工程的关键目标。

每当利益相关者希望了解软件系统某方面的解释时，就会产生解释需求 [14,19]。解释需求具有很强的主观性，并会受到各种混杂因素的影响，例如最终用户对系统的个人体验以及使用情境 [5,10,29]。此外，解释的呈现形式和时机等设计决策，也可能影响整体用户体验所受到的影响程度 [4,10]。考虑到这些因素，需求工程师可以将利益相关者的解释需求转化为正式的解释需求 [18,19]。

2.2 用例

在需求工程中，用例是一种成熟的技术，用于指定功能需求并从外部视角描述系统行为 [17,28]。用例描述参与者与系统之间为实现参与者目标而进行的交互 [25,28]。这不仅用于文档编写，也支持与利益相关者的沟通 [9]。

用例表呈现单个用例的详细流程，是需求规格说明的重要组成部分，例如遵循 Volere模板 [26]的需求规格说明。在本文其余部分中，我们将把结构化的用例表 称为用例。它们不同于高层级的 UML 用例图；后者描绘用户与系统之间的边界，以明确哪些用例属于系统范围之内或之外 [26]。

本文采用 Cockburn基于模板的格式来描述用例，该格式将信息组织到预定义字段中，并同时提供文本和表格两种表示形式 [8,17,25]。模板以用例名称开头，用于简要说明目标；随后是前置条件，用于设定用例开始时的预期状态；以及结束条件，用于定义成功结果后的系统状态。其核心是主成功场景，概述从触发到目标完成的所有流程步骤。扩展可以附加到主场景中的特定步骤，并通过条件描述变化、替代路径或非成功的异常场景 [8]。我们将主场景或扩展部分中的步骤统称为用例步骤。

2.3 相关工作

解释需求： 可解释性需求描述的是系统提供解释，而解释需求则从用户视角表达了对澄清的愿望 [18,19]。因此，需求工程师必须首先了解用户的解释需求，才能制定适当的可解释性需求。Droste 等人 [14] 对日常软件系统的用户开展了一项调查研究，以构建软件解释需求的分类体系。该分类体系将解释需求划分为五个主要类别，并包含若干子类别。**交互** 解释用于指导软件的操作与导航，其需求数量最多。第二常见的需求类别涉及**系统行为**解释，其中包括对算法和意外事件的解释，与 XAI 解释相似。每当术语或系统特定元素不清晰时，用户便会提出有关**领域知识** 的解释需求。最后，一些用户还表示需要解释**隐私与安全** 或**用户界面**。

用例：目前，关于解释需求的研究尚未考虑使用需求工件，例如用例或用户故事。然而，在与之密切相关的可用性领域，已有研究探讨了这些工件的使用。两种值得注意的方法探讨了如何利用它们在软件开发的早期阶段识别可用性需求。

Hornbæk 等人 [20] 在其研究中介绍了用例评估（*UCE*），该方法使用主要源自启发式评估的指南，对用例进行系统检查，以预测潜在的可用性问题。检查由与软件无关的可用性专家进行。他们将用例方法与传统的出声思维测试进行了比较，发现两种方法共同识别出了24%的问题类型。同时，34%的问题类型仅通过出声思维测试识别出来，而42%只能通过*UCE*识别出来。定性数据表明，评估人员认为仅凭用例将系统形象化需要很强的想象力。

Oliveira Júnior 等人还提出了一种相关方法 [24]。他们介绍了 $USARP$，这是一种通过用户画像和可用性指南丰富用户故事，从不同用户群体的视角识别和明确可用性需求的方法。可用性指南描述了会影响软件设计和架构、且能够转化为功能需求的可用性机制。这些机制涵盖系统状态、反馈、警告、取消和多级帮助等方面。实证评估表明，$USARP$ 能够帮助获取更全面、更精准的可用性需求，并使获取过程更加轻松。

这两种方法都展示了需求工件在实现前支持识别与可用性相关的问题或需求方面的潜力。然而，两种方法都没有将系统地识别解释需求作为首要目标加以直接探讨。我们的研究在此研究方向的基础上，探究用例表能否作为引出此类需求的来源，并特别关注系统行为和交互。

3 研究设计

3.1 研究目标和研究问题

我们的目标是考察使用用例表引出解释需求是否合适。为此，我们开展了一项用户研究，请参与者阅读用例描述中的步骤，并标注他们需要解释的步骤。我们提出以下研究问题：

RQ1：用例表在多大程度上能帮助用户表达其解释需求？

RQ2：基于用例表陈述的解释需求有多明确？

RQ3：用例表能够捕捉哪些解释需求类别？

本文探讨用例表是否能帮助用户阐明其解释需求（RQ1），具体而言，参与者能否根据用例表述自己的解释需求。此外，我们还要探讨用例表方法能否产生需求工程师能够理解的解释需求（RQ2）。换言之，我们想了解需求工程师是否能够清楚理解用户表达的需求。此外，我们还希望初步了解用例在支持记录不同类别的解释需求方面效果如何（RQ3），从而评估这项技术是否值得用于满足哪些需求。

3.2 研究过程

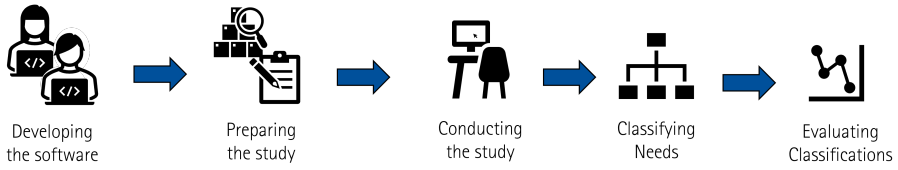


图 1. 我们的方法论概览。

我们的研究方法论总结于图 1 中。首先，我们开发了一款软件应用程序——用例工具 ——以帮助参与者在研究期间记录解释需求。该工具从软件需求规格说明中提取用例并加以展示。对于每个用例步骤，该工具都提供一个文本字段，用户可以在其中输入相应的解释需求（如有）。随后，我们为两个案例应用分别创建了六个用例。

参与者被要求在该工具中仔细查看十二个用例。他们被指示重点关注描述和扩展部分中的步骤。在阅读这些步骤时，参与者需要记下任何产生的解释需求。这项研究共有十三名参与者，每人都具有技术背景。由于所有参与者都接受了关于可解释性领域的简短介绍，因此没有人需要事先了解可解释性。

我们收集了所述的解释需求，并依据 Droste 等人的分类体系 [14] 对其进行分类。两名在可解释性领域经验丰富的编码员独立对这些需求进行分类。对于分类结果中的任何分歧，均由第三位编码员协助解决。

作为评估的一部分，我们分析了解释需求的总体出现情况，并根据编码者间一致性评估参与者所书写需求的明确程度。此外，我们考察了两种应用程序整体以及各个应用程序中每一类别的出现频率。

3.3 参与者的人口统计特征

本研究在 Avacon Netz GmbH 同一工作地点的 13 名员工中开展。我们决定从同一家公司招募参与者。这样，我们就能在研究中使用两个案例应用，而参与者在其工作环境中会接触到这些应用。由此，我们能够确保参与者对这些软件系统具有内在兴趣。此外，所选应用程序与他们的日常工作相关，因此为本研究提供了一个真实的软件使用案例。本研究为自愿参与，且不提供任何报酬。13 名参与者均完成了整个研究，但他们也被告知可以随时退出。10 名参与者自我认定为男性（77%），3 名自我认定为女性（23%）。参与者的年龄在 21 至 59 岁之间。具体而言，46% 的参与者年龄在 30 至 39 岁之间，21 至 29 岁的占 31%，50 至 59 岁的占 23%。就参与者取得的最高学历而言，8 人持有高级技工（“Meister”）或国家认证技师（“Techniker”）资格，3 人获得学士学位，1 人完成了职业培训，另 1 人持有高中毕业文凭（“Abitur”）。

3.4 开发的软件：用例工具

用例工具 允许用户导入软件需求规格说明，并从中提取用例。此外，软件还可以显示从软件需求规格说明中提取的用例图。该工具仅从 PDF文档中提取以表格形式呈现的用例，并在提取完成后显示所有成功提取的用例。用户可以从列表中选择一个用例，以查看完整的用例表。该工具会显示原始用例表中的所有字段，而不仅仅是描述和扩展步骤。在描述和扩展部分的每个步骤旁，用户

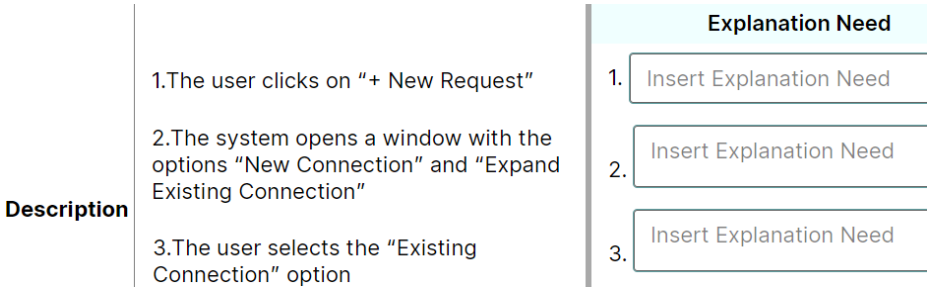


图 2. 工具中显示的用例摘录。

都有一个输入框，用于填写解释需求。图2直观地展示了这一点。标注后的用例可以导出为 PDF 和 CSV 文件。

3.5 准备研究

研究准备工作包括两项活动：为研究选择两个不同的软件应用程序，并为这些应用程序创建所需的用例。所选应用程序与参与者的工作环境相关，因此能够提供更真实的研究场景。我们选择了 *Envilio* 和*Microsoft Teams*（MSTeams）作为应用程序。*Enivilio* 支持电网设计，而*MSTeams* 支持组织任务。后者已广泛应用于参与者的日常工作中，而前者最近才在参与公司引入。

选定应用程序后，我们分别为这两个应用程序创建了六个用例。所有 *Envilio* 用例共包含 67 个用例步骤，而 *MSTeams* 共包含 76 个用例步骤。我们采用 Cockburn 的用例模板 [8] 作为参考。起草的用例涵盖了这两个软件应用程序中从常见到罕见的用例。为匹配员工和所开发工具使用的语言，这些用例均以德语撰写。为确保研究结果可供查阅，我们在补充材料中提供了这些用例及所述解释需求的忠实译文 [2]。

3.6 数据收集

我们收集了两类数据：使用用例工具收集并标注的解释需求，以及通过问卷收集的参与者人口统计数据。研究持续了 40 至 50 分钟。

3.7 数据分析

导出的已注释用例表经过人工分析。首先，两名具有可解释性领域经验的编码员根据 Droste 等人的分类法对解释需求进行分类 [14]。所提供的指南

由 Droste 等人提供的 [13,15] 用于支持在编码过程中更加客观地对解释需求进行分类。编码员可以将任何明确提出的解释需求归类为“无需求”。编码员从我们的数据中识别出的不适用需求示例包括“非常好”或“这应该可以在不使用名称的情况下实现”等评论。完成初始编码后，第三位编码员解决了现有的冲突。随后，计算了总体上的编码者间一致性以及每个软件的编码者间一致性。这些数值有助于评估所收集解释需求的清晰性和明确性。最后，我们检查了总体上以及每个软件中各类别需求的分布情况。

4 结果

我们研究的十三名参与者对十二个用例中的共计165个用例步骤进行了标注。单个标注可能包含若干不同的模糊方面，我们将其称为区段。在这165个标注中，两名编码员识别出了173个不同的区段。其中约20个区段（11.56%）未产生解释需求。相比之下，153个区段（88.44%）产生了解释需求。在我们识别出的173个区段中，113个属于$Envilio$的用例，其中8个区段（7.10%）不包含解释需求。其余60个区段属于$MSTeams$，其中12个（20%）不包含解释需求。

4.1 编码者间一致性

两名编码员根据 Droste 等人的分类法 [14] 对这些区段进行了分类。他们达到了76.28%的编码者间一致性（ICA），并根据 Brennan 和 Prediger [3] 的方法，获得了 $\kappa = 0.72$ 的经机会校正的κ系数。这表明，按照 Landis 和 Koch [22] 的定义，双方达成了较高程度的一致。两个应用程序合计以及各解释需求类别的 ICA 列于表1中。

表 1。 按解释需求类别对两名编码员均进行分类的章节的 ICA。

类别	# 章节总数	ICA	ICA MSTeams	ICA Envilio
系统行为	130	78.46%	80.95%	77.27 %
交互	110	75.45 %	71.79 %	77.46 %
领域知识	53	71.70 %	66.67 %	72.73 %
隐私与安全	17	88.24 %	80.00 %	100.00 %
用户界面	2	0 %	0 %	0 %

将两款应用程序放在一起考察时，系统行为、交互和 领域知识 的 ICA 介于 70 % 和 80 % 之间。在所有类别中，隐私 & 安全 的 ICA 最高，为 88%；而用户界面最低，为 0 %。分别考察这三款应用程序时，前三个类别的数值仍然相对相近。不过，数值差异在

MSTeams 中更大（66 % - 80 %），而*Envilio* 中的差异较小（72 % - 77 %）。差异最大的是隐私 & 安全，其中*MSTeams* 的 ICA 为 80 %，而*Envilio*为 100 %。对于用户界面，两款应用程序的 ICA 均保持在 0 %。其余分歧由第三位编码员解决。如果第三位编码员建议的类别与最初选定的类别不同，双方会进行讨论，以达成共识。

4.2 解释需求的类别

解决冲突后，各类别中的解释需求最终分布情况见表2。各软件中解释需求的类别分布情况见表3和表4。

表2. 解决冲突后各类别中解释需求的分布情况。

类别	# 需求数
系统行为	67
交互	52
领域知识	23
隐私与安全	10
用户界面	1

在我们发现的跨系统的 153 项解释需求中，大多数与系统行为 有关（67 项），其次是交互 （52 项）。与领域知识 有关的需求共 23 项，与隐私 & 安全有关的需求共 10 项。另有一项需求被归入用户界面类别。*MSTeams*共确定了 48 项解释需求，而*Envilio*有 105 项。对于系统行为 和交互，两款应用程序的分布相似，前一类别的需求数量多于后一类别。相比之下，领域知识 在*MSTeams* 中的占比为 10%（5 项），约为*Envilio* 中占比（17%，18 项）的一半。隐私 & 安全相关需求在*MSTeams*中占 12%（6 项），而*Envilio*中仅为其三分之一，即 4%（4 项）。*MSTeams*有一项需求被归入用户界面 类别，而*Envilio*没有需求被归入该类别。

表3。 解决*MSTeams*的冲突后各类别的解释需求分布。

类别	# 需求 %	
系统行为	20	41.7
交互	17	35.4
领域知识	5	10.4
隐私与安全	6	12.5
用户界面	0	0

表4。解决 *Envilio* 的冲突后，各类别解释需求的分布。

类别	# 需求 %	
系统行为	47	44,8
交互	35	33,3
领域知识	18	17,1
隐私与安全	4	3,8
用户界面	1	1,0

5 讨论

5.1 对结果的反思

解释需求数量的差异：总体而言，两款受检应用程序的解释需求数量有所不同。值得注意的是，*Envilio* 的用例表中有约两倍于*MSTeams* 的章节数（113 对 60）。这一差异并非由用例步骤总数造成，因为两款应用程序的用例步骤数量相近。*MSTeams* 的步骤总数反而多出九步，这进一步表明差异并非由步骤数量所致。此外，*MSTeams* 中标记为“无需解释”的章节数约为*Envilio*的 1.5 倍。相反，用户表达的解释需求总数在*Envilio* 中约为*MSTeams*的两倍。一个可能的原因是，*MSTeams* 面向更广泛的利益相关者群体，旨在让各类用户都能使用，因此*MSTeams* 可能比*Envilio*更容易理解。相比之下，*Envilio* 是为特定的专业用途而设计，面向专家用户群体。此外，这一差异也可能与此前较长时间的使用有关。*MSTeams* 已在公司内广泛使用，而*Envilio* 仍处于引入阶段。

与其他获取方法的比较：采用用例表的方法后，在一项持续约 45 分钟的类调查研究中，每位参与者识别出了约 12 项需求。在我们的研究中，参与者可以查阅 Droste 等人开发的分类体系 [14]。因此，我们将研究结果与同样采用该分类体系的研究进行比较，以确保差异并非仅仅源于分类体系的存在。Droste 等人 [13] 通过焦点小组识别解释需求。在这一过程中，平均每人在 60 分钟内识别出了约 10 项需求。Obaidi 等人 [23] 考察了不同的获取方法；在焦点小组中，参与者平均在约 54 分钟内识别出了约 5 项需求。在访谈中，参与者在约 34 分钟内识别出了约 12 项需求。Deters 等人 [12] 的论文采用在线调查来识别解释需求。平均而言，每位参与者报告了约 2 项需求。与问卷调查相比，我们的方法耗时更长，但参与者识别出的需求明显更多。焦点小组耗时更长，而每位参与者识别出的解释需求更少。访谈识别出的需求数量相近，但每位参与者所需的时间更少。不过，访谈还需要一名访谈员参与这一过程，而我们的方法则不需要。

解释需求类别的差异：数据进一步表明，在两个软件系统中，系统行为 和交互 是已识别解释需求数量最多的类别。这与用例的主要目的相一致：描述用户与系统之间的交互。与 Droste 等人 [14], 的研究结果相比，我们在系统行为 中观察到的解释需求多于交互。一种可能的原因

是用例会明确概述系统行为，通常使用“系统执行.....”之类的自然语言陈述。此外，分析显示，领域知识 在$Envilio$ 中的需求占比高于在$MSTeams$中的占比。相比之下，隐私 & 安全 在$MSTeams$ 中的需求占比相对高于在 Envilio 中的占比。这一差异可能归因于$Envilio$是一款面向专家的应用，而$MSTeams$ 面向更广泛的受众。另一种可能是，用户认为$MSTeams$ 会收集更多与隐私相关的数据。用例似乎很少产生用户界面 类别中的解释需求，因为它们没有传达有关用户与用户界面进行交互的细节。

解释需求的明确性：用户标注的部分由两名编码员进行评估。编码员将每个部分归入一个解释需求类别，或标记为“无需求”。ICA 表示两名编码员的分类结果有多大程度保持一致。这可以作为明确性的间接指标。含糊或不清晰陈述的比例较高，可能会导致编码员之间产生分歧。编码类别定义不清也可能导致分歧。然而，该编码方案此前已在研究中成功应用 [16]。根据 Landis 和 Koch [22]，的定义，所达到的实质性一致性表明，解释需求陈述相对清晰且明确。需要注意的是，两名编码员都是具有可解释性研究经验的研究人员，这可能影响了 ICA。由于在用户界面 类别中发现的需求总数较少（2 个），因此应谨慎解读 0% 的 ICA。在所有应用程序中，两名编码员总共仅将两个解释需求归入该类别。然而，两名编码员均未将其中任何一个需求归入该类别。

5.2 回答研究问题

RQ1 – 用例表在多大程度上能帮助用户表达其解释需求？ 参与者能够记录他们对两个应用程序用例中的用例步骤所提出的解释需求。参与者总共记录了 173 项潜在解释需求，其中大多数被编码员归类为有效需求。与其他采用不同引出方法和相同分类体系的研究所报告的数量相比，参与者识别出的需求数量较多，这表明用例表有助于参与者表达解释需求。参与者能够针对成熟的软件（$MSTeams$）和新引入的软件（$Envilio$）报告解释需求。这些发现表明，用例表可以作为获取解释需求的合适手段。

RQ2 – 基于用例表陈述的解释需求有多明确？ 参与者在用例表步骤旁记录的解释需求通常都能被编码员一致理解。编码员在根据 Droste 等人 定义的类别对这些需求进行分类时达成了较高程度的一致性。这表明，这些需求足够清晰明确，能够被一致地解读。

ICA 值约为 72 % 至 88 %，而用户界面 类别是主要的例外，其 ICA 为 0 %。由于需求数量较少，这一 ICA 值不足以为判断该类别的歧义性提供可靠证据。总体而言，结果表明，通过用例表收集的需求通常是易于理解的。

RQ3 – 使用用例表可以捕获哪些解释需求类别? 数量最多的解释需求被归类为系统行为，其次是交互。这两个类别均包含超过 50 条需求。这表明，用例表适合收集这两个类别中的解释需求。通过用例表还收集到了超过 20 条来自领域知识的解释需求，这表明用例表同样可以帮助用户报告这些需求。此外，还发现了来自隐私和安全的需求。鉴于需求数量相对较少，结果可能取决于所研究的应用。最后，仅为用户界面识别出一条解释需求。因此，无法就用例表是否适合捕获该类别中的解释需求得出可靠结论。

5.3 效度威胁

我们遵循 Wohlin 等人 [31]的指南，讨论研究科学效度所面临的威胁：

构念效度：这些威胁涉及所开展的研究是否恰当地反映了所构建的理论。如果所有参与者首先查看同一应用的用例，学习偏差就可能影响我们的研究结果。因此，一半参与者从$Envilio$开始，另一半则从$MSTeams$的用例开始。所有用例均由同一名研究人员编写，因此其写作风格可能会影响研究。为减轻这种影响，另一名研究人员审阅了所有用例。最后，还需要确保参与者对解释需求具有共同理解。为此，我们事先向所有参与者介绍并解释了 Droste 等人 [14]的分类法。

内部效度： 这些威胁与研究结果是否独立于（未知的）第三因素有关。本研究以现有软件为基础设计用例，这可能有助于参与者更好地设想这些用例，从而影响他们记下的解释需求数量。此外，熟悉其中一款软件应用程序可能会让人对任务的执行方式形成固定看法。我们的方法可能容易受到虚假记忆偏差 [27]的影响，导致用户忘记列出的步骤，进而产生更多解释需求。因此，这些解释需求可能源于这种

偏差，而非确实需要解释该步骤。尤其是，*MSTeams* 已在公司内广泛使用，而*Envilio*则不然。因此，解释需求数量上的差异可能源于这一点。

结论效度： 这些威胁与结果和测量值之间的统计相关性有关。编码过程始终会受到编码者对主题的个人见解和理解的影响，因此具有主观性。然而，我们研究中的编码者间一致性较高，这表明我们的分类具有稳健性，并且其他研究人员在遵循相同指南对我们的数据进行编码时，也会得出相近的结果。

外部效度： 这些威胁与将研究结果推广到研究范围之外有关。由于研究仅纳入了来自一个领域的 13 名参与者，并且只考察了两个系统，因此研究结果的可推广性受到限制。此外，研究仅涉及这两个系统中的少数用例。这三类威胁源于本研究旨在提供初步见解这一事实。在参与者数量更多、涵盖不同领域的应用程序更多、用例范围更广的综合性研究中，可以应对这些威胁。

6 结论与未来工作

从我们的研究中获得的初步见解表明，用例可以帮助用户在阅读用例时阐述解释需求。研究中的参与者能够通过使用用例表，清晰地表达不同类型的解释需求。这些结果可以作为深入分析该过程在更现实场景中的适用性及其影响的起点。用例为在软件开发早期引出解释需求提供了机会，从而有助于在系统中及早考虑尚不明确的用户需求。

虽然本文采用了现有应用程序中的用例，但我们计划在未来的研究中，将我们的方法应用于一个正在进行的软件开发项目。通过这种方式，我们希望评估该方法在真实环境中的适用性，并调查在使用软件的过程中，所引出的需求是否确实会出现。此外，我们还希望研究通过用例引出的解释需求，是否与通过原型或其他方法引出的解释需求相似。通过这些研究，我们旨在考察使用用例引出解释需求的有效性，以及该方法在软件开发项目中可能带来的益处。

数据可用性声明 支持本研究的所有材料，包括完整翻译的用例表和解释需求、参与者人口统计数据、由两名编码员分类的解释需求，以及用例工具的代码仓库，均已根据知识共享署名 4.0 国际许可协议在 Figshare 上公开提供，网址为 https://figshare.com/s/922811e47aa0dba995be。复现包包含重现我们研究结果所需的全部材料。

致谢。 本研究由德国研究基金会（DFG，德国研究基金会）资助，资助编号：470146331，项目为 softXplainer（2026—2029 年）。本文部分内容使用了 DeepL 以改进语言表达。

利益披露。作者声明不存在与本文内容相关的利益冲突。

参考文献

1. Adadi, A., Berrada, M.：窥探黑盒：关于可解释人工智能（XAI）的综述。IEEE Access **6**，52138–52160（2018） 2. Anonymous：补充材料——“用例表如何支持解释需求的获取”（REFSQ’27），2026年11月，补充材料在线获取地址：https://figshare.com/s/922811e47aa0dba995be 3. Brennan, R.L., Prediger, D.J.：Kappa系数：一些用途、误用及替代方案。Educational and Psychological Measurement **41**（3），687–699（1981）。https://doi.org/10.1177/001316448104100307，https://doi.org/10.1177/001316448104100307 4. Brunotte, W., Specht, A., Chazette, L., Schneider, K.：隐私解释——获得终端用户信任的一种手段。Journal of Systems and Software **195**，111545（2023） 5. Chazette, L., Brunotte, W., Speith, T.：探索可解释性：定义、模型与知识目录。载于：2021 IEEE第29届国际需求工程会议（RE）。第197–208页。IEEE（2021） 6. Chazette, L., Klünder, J., Balci, M., Schneider, K.：我们如何开发可解释系统？来自文献综述和访谈研究的洞见。载于：国际软件与系统过程会议及国际全球软件工程会议论文集。第1–12页（2022） 7. Chazette, L., Schneider, K.：将可解释性作为非功能需求：挑战与建议。Requirements Engineering **25**（4），493–514（2020） 8. Cockburn, Allister：基本用例模板。技术报告，Humans and Technology（1998） 9. Dalpiaz, F., Sturm, A.：使用用户故事和用例进行需求概念化：一项对照实验。载于：国际需求工程工作会议：软件质量基础。第221–238页。施普林格（2020） 10. Deters, H., Droste, J., Hess, A., Klös, V., Schneider, K., Speith, T., Vogelsang, A.：X因素：用户体验与可解释性之间的关系。载于：第13届北欧人机交互会议论文集。第1–12页（2024） 11. Deters, H., Droste, J., Schneider, K.：洞察需求获取的脉搏：生理触发因素与可解释性需求。载于：REFSQ研讨会（2024） 12. Deters, H., Reinhardt, L., Droste, J., Obaidi, M., Schneider, K.：识别解释需求：迈向基于用户的指标目录。载于：2025 IEEE第33届国际需求工程会议（RE）。第31–42页（2025）。https://doi.org/10 00014 13 1109 63999 2025 . /RE . . . Droste, J., Deters, H., Obaidi, M., Klünder, J., Schneider, K.：界定可解释的内容——可解释性需求的操作性分类法。Requirements Engineering **30**（2），195–217（2025） 13. Droste, J., Deters, H., Obaidi, M., Schneider, K.：日常软件系统中的解释：迈向可解释性需求分类体系。载于：2024 IEEE第32届国际需求工程会议（RE）。第55–66页。IEEE（2024）

15. Droste, J., Deters, H., Obaidi, M., Schneider, K.：补充材料——“日常软件系统中的解释：迈向可解释性需求分类体系”（RE’24 10871086 16 2024）（2026年3月），在线获取地址：https://zenodo.org/records/ 。Droste, J., Wellhausen, C., Deters, H., Schüring, B., Specht, A., Schneider, K.：在线探查群体：对子版块帖子中解释需求的半自动化分析。载于：2026 IEEE第34届国际需求工程会议研讨会（REW）。第127–133页。IEEE（2026） 16. Frattini, J., Frattini, A.：采用用例描述进行需求规格说明：一项工业案例研究。载于：2025 IEEE第33届国际需求工程会议（RE）。第179–191页。IEEE（2025） 17. Groen, E.C., Deters, H., Droste, J., Schmitt, H.：横切质量属性需求：集成可解释性与隐私，迈向利益相关者需求分类体系的第一步。RE Magazine（IREB）（2026），在线获取地址：https://ireb.org/de/re-mag/requirements-for-cross-cutting-qualities 18. Groen, E.C., Deters, H., Droste, J., Schmitt, H.：利益相关者需求作为横切质量属性的新需求工程制品。载于：Softwaretechnik-Trends第46卷第3期。德国信息学会（2026） 19. Hornbæk, K., Høegh, R.T., Pedersen, M.B., Stage, J.：用例评估（UCE）：一种用于软件开发早期可用性评估的方法。载于：IFIP人机交互会议。第578–591页。施普林格（2007） 20. Kieras, D.E., Bovair, S.：心智模型在学习操作设备中的作用。认知科学 **8**（3），255–273（1984） 21. Landis, J.R., Koch, G.G.：分类数据观察者一致性的测量。Biometrics **33**，159–174（1977） 22. Obaidi, M., Droste, J., Deters, H., Herrmann, M., Ochsner, R., Schneider, K., Klün- der, J.：如何获取可解释性需求？访谈、焦点小组和问卷调查的比较。载于：2025 IEEE第33届国际需求工程会议（RE）。第167–178页（2025）。https://doi.org/10.1109/RE63999.2025. 00025 23. de Oliveira, G.F., Ferreira, B., Marques, A.B.：Usarp方法：利用用户画像和用户故事获取并描述可用性需求。载于：第34届巴西软件工程研讨会论文集。第437–446页（2020） 24. Pohl, K.：需求工程：基础、原则与技术。施普林格柏林海德堡，第2版（2025）。https://doi.org/10.1007/ 978-3-662-69205-9 25. Robertson, J., Robertson, S.：Volere。需求规格说明模板（2000） 26. Seghezzi, S., Armstrong, M.J., Haggard, P.：“本可能发生什么”的代价：关于反事实行动的虚假记忆。认知 **278**，106701（2027年）。https://doi.org/https://doi.org/10.1016/j.cognition.2026.106701，https: // www.sciencedirect.com/science/article/pii/S0010027726002702 27. Sinnig, D., Chalin, P., Khendek, F.：用例与任务模型：一种集成开发方法及其形式化基础。ACM软件工程与方法学汇刊（TOSEM）**22**（3），1–31（2013） 28. Sokol, K., Flach, P.：一种解释并不适用于所有人：交互式解释对机器学习透明度的前景。KI-人工智能 **34**（2），235–250（2020） 29. Vigano, L., Magazzeni, D.：可解释安全。载于：2020 IEEE欧洲安全与隐私研讨会（EuroS&PW）。第293–300页。IEEE（2020） 30. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., Wesslén, A.：软件工程中的实验。施普林格；海德堡（2012）