arXiv:2304.02702v1 [cs.SE] 5 Apr 2023

需求工程中的人为错误管理：我们应该改进人员、流程还是环境?

Sweta Mahaju[a]、Jeffrey C. Carver[b]、Gary L. Bradshaw[c]

[a]计算机科学系，中田纳西州立大学，默弗里斯伯勒，田纳西州，美国[b]计算机科学系，阿拉巴马大学，塔斯卡卢萨，阿拉巴马州，美国[c]心理学系，密西西比州立大学，斯塔克维尔，密西西比州，美国

摘要

背景：软件开发以人为中心，因此容易受到人为错误的影响。人为错误是人类思维过程中的错误。为了确保软件质量，从业者必须了解如何管理这些人为错误。组织通常会对需求工程过程进行调整，以预防人为错误的发生，或在错误确实发生时缓解其造成的危害。尽管其他学科已有关于人为错误管理的研究，但软件工程，尤其是需求工程中有关人为错误预防与缓解的研究仍然有限。目前的软件工程研究尚未就需求工程中哪些类型的调整最为有效给出有力结论。目标：本文旨在基于从需求工程专业人员处收集的数据，构建人为错误预防与缓解策略分类体系。方法：我们对两项关于需求工程实践的从业者调查数据进行了定性分析，以识别并分类人为错误预防与缓解策略。结果：我们根据调整主要影响人员、流程还是环境，将人为错误管理策略整理为一个分类体系。在这些高层类别中，我们进一步将策略划分为低层类别。结果显示，超过50%的已报告策略需要调整流程，23%需要调整环境，21%需要调整人员，剩余5%因过于模糊而无法分类。此外，超过50%的策略侧重于需求工程中的管理活动。结论：人为错误管理分类体系为需求工程中人为错误预防与缓解策略提供了系统的分类与组织方式。

提交至 *Information and Software Technology* 的预印本 *2023年4月7日*

这种系统化的组织方式为后续研究奠定了基础。

关键词：人为错误, 认知心理学, 错误预防,错误缓解, 需求工程

---

1. 引言

需求工程是软件开发生命周期（SDLC）的基础阶段。在此阶段，人员相互交互以引出、记录和验证软件需求。此阶段人员的参与为人为错误提供了大量机会。人为错误 是人类思维过程中的失败，如果未得到处理，可能导致需求文档中的故障。当需求工程师误解客户需求、应用错误类型的解决方案，或因记忆超载而犯错时，就会发生人为错误 [1]。这些类型的人为错误特别成问题，因为它们的影响可能长期隐藏，并且往往只在软件开发过程的后期才以失败的形式显现 [2]。因此，频繁的返工和需求变更可能导致更多错误、降低软件质量并增加成本 [3]。

当发生人为错误或任何其他类型的次优流程时，团队必须采取措施缓解错误的潜在影响，或引入变更以预防未来类似的错误。尽管该主题对于提高软件质量很重要，但我们未能找到关于需求工程中这些类型的预防和缓解技术的已发表系统研究。为了解决文献中的这一不足，我们需要回答如下问题：我们如何将预防和缓解方法组织成一个揭示其底层结构的框架？以及我们能否识别出可能有效或无效的特定类型的预防和缓解方法？这些问题的答案将帮助软件开发组织主动实施适当的错误预防与缓解技术，并避免从自身错误中学习的痛苦过程。对更广泛文献的回顾还表明，某些类型的错误缓解和预防策略不太可能有效，而其他技术更有可能被证明有效。

人为错误管理通过开展错误预防和错误缓解活动来提升软件质量。错误预防活动通过改进开发过程来降低发生错误的可能性。错误缓解

相关活动旨在尽量减少错误发生后对下游造成的影响。然而，鉴于缺乏关于错误管理方法的研究，人为错误在需求工程中仍然持续存在。尽管如此，即使没有已发表研究分析有效的错误预防与缓解策略，个体开发人员仍一直在努力预防错误。因此，研究人员和从业者都可以从对错误预防与缓解策略有效性的正式分析中获益。此外，对错误预防与缓解策略进行分类，将为未来研究奠定基础。因此，本文的主要研究目标是构建人为错误管理分类体系，并对需求工程过程各阶段的错误预防与缓解策略进行分类。

为实现这一目标，我们分析了对专业需求工程师开展的调查数据 [4, 5]。这些调查收集了需求工程流程中遇到的问题，以及管理这些问题的方法 [5]。在先前的研究中，两位作者分析了这些数据，以识别从业者为应对不同类型的人为错误而采用的具体错误预防与缓解方法 [4]。该研究旨在提取针对每种错误的特定预防或缓解策略，并将每项策略标记为预防 缓解 或两者兼具。

在本研究中，我们对先前研究中提取的预防与缓解策略进行了更深入的分析。先前研究的成果构成了本文所述研究使用的“原始数据”。先前研究列出了各项策略，并说明它们主要用于预防还是缓解，但并未进一步深入分析这些策略并对其进行系统化归类。本文延续了这项研究，系统地构建了预防与缓解策略的分类体系，并指出这些策略最适用的需求活动。为落实这一方法，我们按照 Kitchenham 等人 [6] 的建议，采用了由个人分析与编码、继而开展讨论组成的流程。因此，本研究实现了构建有序的错误管理策略结构这一目标。

本文的主要贡献如下：

1. 对行业中使用的人为错误预防与缓解策略进行分析，2. 分析这些活动中的哪些发生在每项需求工-程活动中，以及 3. 将这些策略分类为一个包含其需求工程活动的分类体系。

2. 研究背景

本节介绍有关人为错误，以及软件工程和其他领域中的错误预防与缓解的背景研究。

2.1. 人为错误研究

一种人为错误是指人类思维过程中出现的错误，它会导致行为偏离预期，从而产生意料之外或不理想的结果 [1]。人们在日常生活中都会经历人为错误。几个简单的例子包括：（1）走进一个房间后忘记自己为什么进来；（2）把鸡蛋扔进垃圾桶，而不是把蛋壳扔进去；（3）眼镜明明戴在头上，却还在四处寻找。在需求工程中，如第 1 节所述，人为错误是由需求工程师与顾客之间的误解或沟通不畅造成的。需求工程中的人为错误包括打字错误、需求遗漏、信息缺失以及错误假设 [7]。

其他领域关于人为错误的既有研究表明，试图通过鼓励相关人员改变自身来解决人为错误，并非有效的 [8, 9, 10] 方法。例如，Kellogg 等人历时八年调查并分析了医疗保健领域人为错误的根源 [9]。他们确定并研究了应对人为错误最常见解决方案的有效性。他们得出结论，着眼于改变人员的解决方案（例如提醒他们不要犯错）鲜少有效，因为促使错误发生的情境依然存在。我们预计，公司和组织在需求工程中实施预防与缓解措施时，也会出现相同的结果。例如，人员流动时，经过培训以预防错误的需求工程师离开公司后，接替者可能会重蹈覆辙。

Kellogg 等人进一步强调，改变技术、工具或组织政策的错误管理方法更有用，但人们通常不会提出这类解决方案 [9]。第二次世界大战期间灾难性的“起落架事故”就是这一论点的一个例证。二战前两年发生了 2000 多起飞机事故，原因是飞机着陆时，机长把襟翼控制装置误认为起落架控制装置。飞机在起落架仍收起的情况下着陆，往往造成严重损坏。后续调查发现，发生人为错误是因为起落架控制杆和襟翼控制杆的形状、大小和操作方式完全相同，而且位置相邻。作为一种缓解措施，在重新设计起落架和控制杆之后，

事故因此有所减少。这个例子表明，即使经过正确训练的飞行员，在恶劣条件下也容易发生人为错误 [11]。人为错误专家意识到，与其训练人员操作设计不佳的机器并寄望于取得最佳结果，不如专注于设计更好的机器。同样，需求工程中许多作为问题解决策略而实施的变更采用了无效措施，可能无法帮助解决问题。

2.2. 需求工程中的人为错误研究

为帮助研究人员和从业者更好地理解需求工程过程中出现的人为错误类型，Anu 等人开展了系统性文献综述，以识别、整理并构建文献中描述的人为错误分类体系 [7]。基于认知心理学中的标准人为错误分类体系 [1], , 他们创建了人为错误分类法（HET），用于整理需求工程中的人为错误。HET 扩展了 Reason 提出的疏忽、遗漏和错误等人为错误类别，并提供了一个框架，用于描述需求工程师可能犯下的人为错误类型。了解各种人为错误类型，有助于深入认识最适当的预防和缓解方法。

2.3. 软件工程中的错误预防与缓解方法

Huang 等人通过调查中国航空软件行业的软件工程师所犯的人为错误，制定了人为错误分类法 [12]。他们制定该分类法，旨在帮助预防软件工程中的人为错误。作为研究成果，作者提出了 DPeHE（基于人为错误机制的缺陷预防方法）框架 [13]。DPeHE 由两个模块组成，旨在提升从业者的元认知能力：知识培训和调节培训。知识培训模块可以帮助从业者熟悉他们会犯为什么、如何以及什么 类型的错误。调节培训模块则可以帮助从业者运用根本原因识别方法开展问题解决活动，该方法基于通过知识培训获得的知识。一系列案例研究表明，通过提升软件开发人员的元认知能力，DPeHE 框架在预防错误方面具有有效性和实用价值。然而，由于这项研究仅在两家公司中评估了该框架，其应用范围和有效性是否具有普遍性仍存疑问。

Huang 还研究了完成后活动中的人为错误 [14]。完成后错误是指从业者完成主要任务后，未能完成某个子任务而产生的错误。完成后错误的一个例子是，使用 ATM（自动柜员机）完成交易后忘记取回银行卡 [14, 15]。Huang 在一场编程竞赛中开展了一项实验，其中包含一项需要在主要任务结束时完成的子任务。在参与研究的 55 名学生中，有 23 人遗漏了该子任务。她发现，尽管学生有动力且接受过避免人为错误的培训，他们仍容易犯错。她还强调，人员相关解决方案无法有效应对人为错误。Huang 提出了三种防范完成后错误的策略：(1) 消除不必要的子任务；(2) 改变任务流程；或 (3) 用醒目的颜色或粗体字突出显示该任务。这些策略都侧重于改变流程或程序。尽管这项研究的结果很重要，但参与者是学生而非从业者，这可能会降低其外部效度。

Firesmith 识别了最常见的需求工程问题，解释了这些问题的影响，并提出了可能的解决方案 [16]。作者描述了这些需求问题及其成因，从而阐明了这些问题背后的人为错误。该论文的主要目标是为处理需求问题提供指导。尽管具备人为错误相关知识的读者可能会理解其中对所提出的人为错误管理方法的隐含解释，这些方法能够帮助他们做好应对人为错误的准备，但论文本身并未具体描述人为错误。因此，就人为错误而言，该论文的局限性在于，它依赖于从业者从论文所解释的需求问题中识别人为错误的能力。此外，该研究缺乏坚实的认知视角，且基于作者的经验和观点。

Lopes 等人借助需求工程师 [17]，分析了错误类型在需求问题背景下的重要性。他们开发了一个专家系统，为管理需求错误提供解决方案。该系统接收需求问题列表，并提供一份按重要性排序的人为错误类型列表，这些错误可能已经发生，同时还提供一项源自文献的通用解决方案建议。由于该专家系统使从业者意识到导致需求问题的错误，因此它既是错误预防工具，也是错误缓解工具。然而，该系统的主要局限在于，所提供的解决方案来自文献而非实践，这引发了人们对这些方案在实践中有效性的疑问。

Walia 和 Carver 开展了一项系统文献综述，以识别并分类软件工程需求阶段发生的人为错——

误 [18]。他们利用这些信息构建了需求错误分类体系（RET）。随后，借助 RET 中的错误类型，他们开展了一项实证研究，使用 RET 增强 Lanubile 等人的错误抽象方法 [19]，以预防需求阶段的错误 [20]。错误抽象是一种回溯性流程，即将已识别的故障追溯至其根本错误，并利用这些已识别的错误发现更多故障。研究结果表明，在需求检查期间发现更多错误的参与者，在后续需求创建过程中犯下的错误更少，这说明 RET 可用作错误预防工具 [20]。尽管该研究支持了 RET 作为错误预防工具的有效性，但 RET 本身并非建立在坚实的人为错误理论之上。

Hu 等人利用人为错误分类法（HET） [4, 7]中的错误信息实施了错误抽象流程。他们收集了行业中遇到的需求问题相关数据，将这些问题映射到相应的人为错误，并对需求工程师用于解决问题的方案进行分类。他们将每种方案归类为预防方法（即防止错误发生）或缓解方法（即在错误发生时管理其影响）。随后，他们将每种方案映射到相应的错误类型。虽然这项研究有助于从业者更加了解预防与缓解策略，但针对每种错误类型识别适当策略的能力仍取决于从业者自身的能力。

这一讨论表明，仍有必要开展更多研究，专门探讨需求工程中的人为错误预防与缓解，或更广泛地探讨软件工程中的这一问题。现有研究仍存在尚未解决的空白，包括缺乏坚实的认知理论基础、普适性与有效性问题、依赖开发人员识别正确错误管理策略的能力，以及缺乏对错误预防与缓解机制的系统化组织。本研究旨在弥补这些空白。

3. 研究方法

本节描述了研究的动机、研究问题、数据收集程序以及分析方法的概述。由于分析是迭代的，每一步都建立在前一步的结果之上，我们在第4节中结合结果描述分析的细节。

3.1. 新研究动机

前期工作（第 2.3 节）介绍了对人为错误预防与缓解相关数据的初步分析。这项初步工作仍留下了许多尚未解决、需要进一步研究的空白。具体而言，目前缺乏指导，无法帮助从业者识别针对特定需求问题最合适的预防或缓解策略。

在本文中，我们使用与之前研究相同的工业数据集（如第 3.3.1节和第 3.3.2节所述）进行更深入的分析，将这些研究中确定的预防与缓解策略整理为一个包含详细类别和子类别的分类体系。我们还考察了每项策略在哪些具体的需求工程活动中有用，以便为从业者提供额外的信息。

3.2. 研究问题

以下研究问题指导本研究：

RQ1 从业者在需求工程中采用哪些预防与缓解策略来应对人为错误，以及如何将这些策略组织成形式化分类体系？

RQ2 我们能否将人为错误预防与缓解策略归入 RQ1 中构建的分类体系？

3.3. 研究数据

为了开展分析，我们使用了两个现有数据集中的数据，这些数据集包含来自工业开发人员 [4] 的需求工程信息。这些数据集包含从业者所采用的人为错误管理策略相关信息。我们合并了这两个数据集，以聚焦需求工程师在项目中遇到的问题及其解决方案。以下各小节将进一步介绍每个数据集。

3.3.1. 数据集 *1*：*NaPiRE*

NaPiRE（识别需求工程中的痛点）是一项全球倡议，旨在研究和了解需求工程问题的现状。作为这项倡议的一部分，一个国际需求工程研究人员团体每两年开展一次关于行业需求工程实践的调查。这项调查的结果构成了需求工程问题的知识库 [5]。

此处我们使用 2016年调查的数据，因为这是最后一次受访者描述其处理需求问题策略的调查。该调查共有来自 10 个国家、226 家公司的 226 位受访者。虽然调查包含若干问题，但我们的研究仅关注其中一道问题的答案。调查向参与者列出了 21 个常见需求问题（在此前的 NaPiRE 调查中识别），并要求他们选出最常见的 5 个问题，这些问题需符合其环境 [5]。附录 A 列出了这些需求问题。对于每个问题，受访者都列出了用于应对问题的原因、影响和管理策略。由于其中一些需求问题与人为错误有关，这些回答为人为错误的预防与缓解策略提供了参考。

3.3.2. 数据集 *2*：*CAPS*

CAPS（先进公共安全中心）是阿拉巴马大学的一家跨学科研究组织，致力于开发公共安全应用软件。CAPS 为收集需求工程中的人为错误相关信息提供了良好的环境，因为其软件开发团队包括需求工程师、开发人员和业务分析师。在此前的研究中，两位作者对七名需求工程师开展了两阶段调查，以收集他们面临的问题（其中一些与人为错误有关）以及他们用来应对这些问题的策略信息 [4]。表 1 列出了调查问题。

在第一次调查中，解释人为错误的概念后，研究人员提出了表 1 中调查 1 的问题，以便受访者描述需求问题以及他们如何处理这些问题。接下来，在第二次调查之前，作者介绍了人为错误分类法（HET）、其中的错误类型及示例 [7]。我们与参与者面对面交流，向他们介绍 HET。在这次交流中，我们详细解释了分类体系中的每个高层级和低层级错误类别 [4]。随后，研究人员向参与者提出了表 1 中调查 2 的问题，以收集受访者在了解 HET 后能够识别出的其他需求问题。

3.4. 分析方法概述

我们的分析由三个阶段组成：(1) 预防与缓解策略类型的识别，(2) 预防与缓解策略的分类，以及 (3) 分歧讨论。

我们依次执行了前两个阶段。第一阶段的目标是对数据集进行整体分析，识别所报告策略中的模式，并

表 1：CAPS 调查问题 [4]

调查	问题
1	在过去的软件开发流程中，您遇到过哪些类型的需求问题? 过去的软件开发流程?
	针对这些问题，您采用过哪些类型的预防策略? 采取或计划采取哪些措施来解决这些问题?
2	根据我们介绍的人为错误分类法，是否存在广告- 你遇到但未报告的其他人为错误 在第一次调查期间?
	针对上一题所述的错误，哪些类型的预防性..... 您已采取或计划采取以应对这些问题的干预策略- 问题?

最后提取错误管理类别。每位研究人员分别审阅数据集中的策略，以确定各自的类别集合。在这一阶段，我们没有尝试对每项策略进行分类，而只是确定总体类别，以便之后将单项策略归入其中。为确定这些类别，我们每个人都独立识别各种方法之间的共性，从而为策略确定各自的高层级分组。经过讨论并达成一致（见下文第3 阶段）后，我们形成了一套经共同认可的高层级和低层类别。随后，工作进入第二阶段，其主要目标是将每项策-略分别归入第 1阶段确定的类别中。每位研究人员分别分析每项预防与缓解策略，并将其归入第一阶段确定的错误管理类别和子类别中。

第三阶段（分歧讨论）在前两个阶段中多次进行。每次分类活动结束后，我们都会讨论并解决分歧。这些定期讨论有助于确保我们对分类方法的理解一致且正确。每次讨论结束时，所有作者都达成了一致。

图1展示了我们将预防与缓解策略划分为高层级类别、低层级分类和需求活动的方法。由于每一步的方法都建立在前一步的结果之上，第4节将各步骤的详细说明与每一步的结果结合起来。

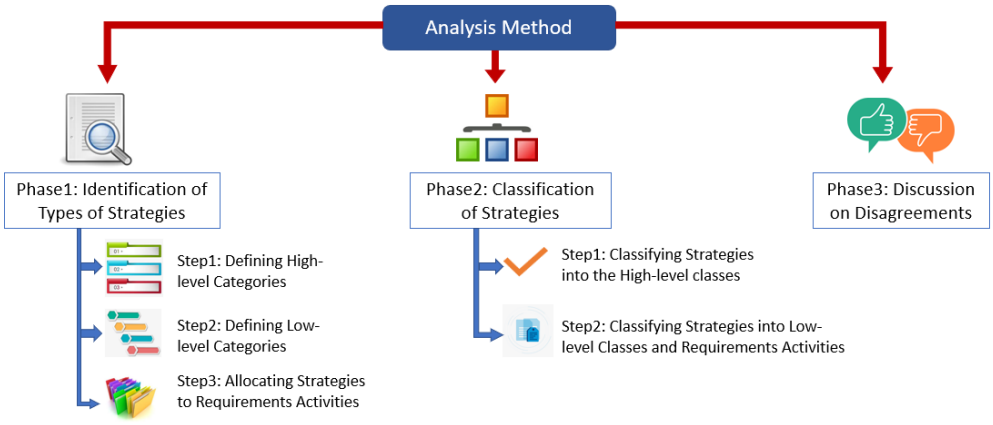


图1：预防与缓解策略的分类

4. 结果

本节首先概述我们用于清理和准备数据以进行分析的流程。随后，本节描述分析的每个阶段的分析流程和结果。我们围绕第 3.2 节提出的研究问题组织本节内容。

4.1. 数据清理

在分析流程中，我们注意到，许多报告中的预防与缓解策略缺乏我们所期望的详细程度。为弥补细节上的不足，我们定义了分类标准。我们观察到，其中一些预防与缓解策略包含两个步骤。例如，策略“引入并使用需求量化方法”包含“引入”和“使用”两个部分，这属于两种不同类型的变更。根据我们的分类标准，当策略包含两个步骤（如引入和使用）时，我们会将该策略拆分为两个策略。因此，我们共有162项预防与缓解策略需要分析。

4.2. RQ1：从业者采用哪些类型的预防与缓解策略来应对需求工程中的人为错误，以及我们如何将其组织成一个形式化分类体系？

为了回答这个问题，我们进行了分析流程的**阶段1：预防与缓解策略类型的识别**。此阶段的主要目标是识别可将预防与缓解策略分类到其中的类别。阶段1有三个相互构建的步骤。我们在下面详细说明这些步骤及其结果。

步骤*1*：定义预防与缓解策略的高层类别

我们首先尝试识别一小套高层类别，以便将预防与缓解方法归入其中。每位作者单独分析数据，以识别所观察到的预防与缓解策略的初始高层类别集。接下来，基于对这些个人分析的讨论，我们商定了高层类别。然后第一作者重新检查个人分析的结果，将其组织到高层类别中并澄清这些类别的定义。接下来，另外两位作者审查了高层类别的定义以及初始结果向这些类别的组织，并记录任何分歧。经过第二次讨论后，我们商定并最终确定了以下高层类别：

1. **人员**：以改变人员行为为主要目标而引入的变更。这些变更包括培训活动、辅导和提醒等。请注意，人员变更会直接影响特定个体。因此，员工流动时，人员变更通常会失效，必须为新员工重新实施。

2. **流程**：直接对任务流程或任务程序所做的更改。过程变化的一个示例是引入审查阶段，要求某些员工在文档进入下一阶段之前签字确认。与人员变化不同，由于流程变化会影响所有员工所遵循的流程，因此流程变化应当能够经受员工流动的影响。流程变化可能发生在管理层面（即管理人员进行更多监督），也可能发生在需求工程层面（即在需求工程过程中新增一个步骤）。

3. **环境**：对需求工程过程所处环境的变更。环境变更的示例包括引入新的开发工具或流程、增加人员

拥有专业知识，或对工作文化进行改变。环境变化通常会导致过程变化。例如，使用新的错误跟踪系统改变了工作环境，也会改变开发流程。

4. **模糊**：此类别涵盖无法明确归入上述某一类别的变更。许多此类模糊变更源于描述缺少必要的细节，使我们无法明确判定变更的类型，以及它是否属于上述某一类别。

步骤 *2*：为预防与缓解策略定义低层类别

定义高层类别后，第一作者再次审阅预防与缓解策略，并补充低层类别，以描述每个高层类别中发生的具体变更类型。另两位作者审阅了低层类别及其在高层类别中的归属，并记录了任何分歧或需要进一步讨论的事项。讨论后，作者们就以下低层类别达成一致。

1. **人员**

(a) **内部培训活动 (ITA)**：通过改变人员来提升其技能和绩效，例如沟通技能或时间管理技能。这种改变以正式方式作用于内部团队成员。例如，规划并开展培训（以提升技能和绩效）(b) **内部团队激励 (ITE)**：通过激励团队、为其提供指导，并让团队成员了解需要做什么以及如何正确完成，来改变人员。这是对内部团队成员进行的非正式指导。例如，激励团队或赋予团队权力；指导团队成员专注于客户需求 (c) **外部培训和辅导活动 (ETA)**：通过提高外部利益相关者的意识或为其提供指导，促使人员发生改变。这种改变以非正式方式作用于团队外部人员，例如客户和顾客。例如，指导客户了解开发人员不一定具备领域知识；培训/辅导/咨询：顾客需要接受

指导，了解软件开发这一阶段的重要性，以及若不这样做，在支持和维护方面会产生多大成本。

2. **流程**

(a) **调查（INV）**：通过识别在流程中应做什么以及如何去做来解决流程中出现的问题，从而实现流程变更。例如，评估工具或识别流程缺口。这并不是确切的流程变更，但调查可能会在之后引起流程变更。调查本身不会解决问题，但有助于尝试确定应采取什么措施来解决问题。(b) **优化与重新设计（RR）**：由以下因素引起的流程变更：a)由于引入新流程、工具或方法而导致的环境变更；b) 根据需要优化或修改现有流程；或c) 将现有流程作为被修改流程的一部分加以实施。例如——实施变更管理流程、制作原型和线框图，以帮助开发人员理解需求，并使用需求量化方法。(c) **角色与职责定义（DR）**：通过更清晰地定义现有角色与职责而发生的流程变更。例如，明确界定角色与职责。(d) **资源分配（RA）**：通过在正确的事件中分配正确的资源而实现的流程变更。例如，在该软件开发项目的需求收集流程中，让代表每个角色的所有人员都参与进来。

3. **环境**

(a) **新增流程、工具或方法（ANP）**：通过在软件开发中引入新的方法、工具或流程而导致的环境变化。例如，引入敏捷方法或新的质量保证流程 (b) **新增人员（AP）**：通过增加通常不参与该子流程工作的人员而导致的环境变化。例如需求专家、最终用户、客户。AP 子类别进一步分为两类：a 内部（Int）：增加为执行需求活动的组织工作的人员。例如，开发人员、管理人员、需求工程师、业务分析师等。

b 外部（Ext）：增加不属于开展需求工程活动的组织的人员。例如，最终用户、客户、外部需求专家。(c) **增加结构（AS）**：通过制定或创建标准或模板来改变环境，以适应特定流程。例如，制定质量标准或创建规格说明模板。(d)**增加新角色（AR）**：通过在现有团队成员中定义适当的角色和职责来改变环境。例如，任命团队负责人或引入经理。

步骤 *3*：将低层类别分配到需求工程生命周期的各个阶段

在步骤 2 的讨论中，我们注意到，预防与缓解策略的描述表明，它们适用于需求生命周期的不同阶段。为体现这一差异，我们进一步根据这些策略所处的需求生命周期阶段，对预防与缓解策略进行了分类。第一作者根据每个低级类别最有可能出现的时机，将步骤 2 中的每个低级类别归入Kotonya和Sommerville [21] 所描述的一项或多项需求活动（见表 2）。其他作者审查了这项归类，讨论了所有分歧，并达成完全一致。

我们将已识别的高层级类别和低层级子类别整理到人为错误管理分类体系（HEMT）中，如图 2 所示。

4.3. RQ2：我们能否将人为错误预防和缓解策略归入 *RQ1* 中构建的分类体系？

为回答这一问题，我们开展了分析流程中的**第2阶段：预防与缓解策略分类**。该阶段的目标是将每项预防与缓解策略归入第1阶段所定义的高层类别、低层类别以及需求活动。第2阶段包含三个相互衔接的步骤。我们在下文详述这些步骤及其结果。

步骤*1*：将预防与缓解策略归入高层级

我们分别将每项预防与缓解策略分类到第1阶段第1步中定义的高层级中。为便于流程，我们随机将162项预防与缓解策略分配到五个批次，分别为40、31、36、40和15项策略。对于最后一个批次，我们添加了13项随机选择的

表 2：软件开发中的需求活动 [4]

需求活动	**描述**
需求获取	包括需求收集在内的活动，处于- 站立、组织以及与客户互动
需求分析	包括需求分类的活动， 优先级排序，以及解决任何误解
需求规格说明	包括定义用户和系统需- 需求与实施，以及测试方法和 验证实现。 它包括一项活动 记录在.....期间收集的信息 将获取和分析活动整理成一份文件，该文件 定义一组需求（系统规格说明 需求文档）。
需求验证	包括对以下内容进行核查和验证的活动 规格说明活动中规定的需求。 歧义性、一致性和冗余性检查。
需求管理	包括管理计划、资源 收集与分配，以及对其中变更的控制 系统需求等等。

预防与缓解策略，这些是我们在之前批次中已审查过的。这些策略第一次都存在分歧，因此我们想验证那些策略的早期分类。在分别对每个批次中的策略进行分类后，我们比较了结果并讨论任何分歧以解决它们。我们在第一批次中识别出17个分歧，第二批次13个，第三批次7个，第四批次8个，第五批次8个。讨论分歧后，三位作者都同意最终分类。此时，我们合并了两项策略。我们还将八项策略分类为模糊，并未考虑它们进行进一步分类。这些更改留下了153项策略用于下一步。

表 3 展示了各项需求活动中，预防与缓解策略在高层级类别和低层级子类别之间的分布，以回答研究问题。按照上述方法，我们首先将预防与缓解策略划分为针对 RQ1 定义的人员、流 程和环境 类别。在这一过程中，我们发现，大多数（51%）预防与缓解策略属于流 程 变更。人员 变更和环境 变更分别占 21% 和 23%

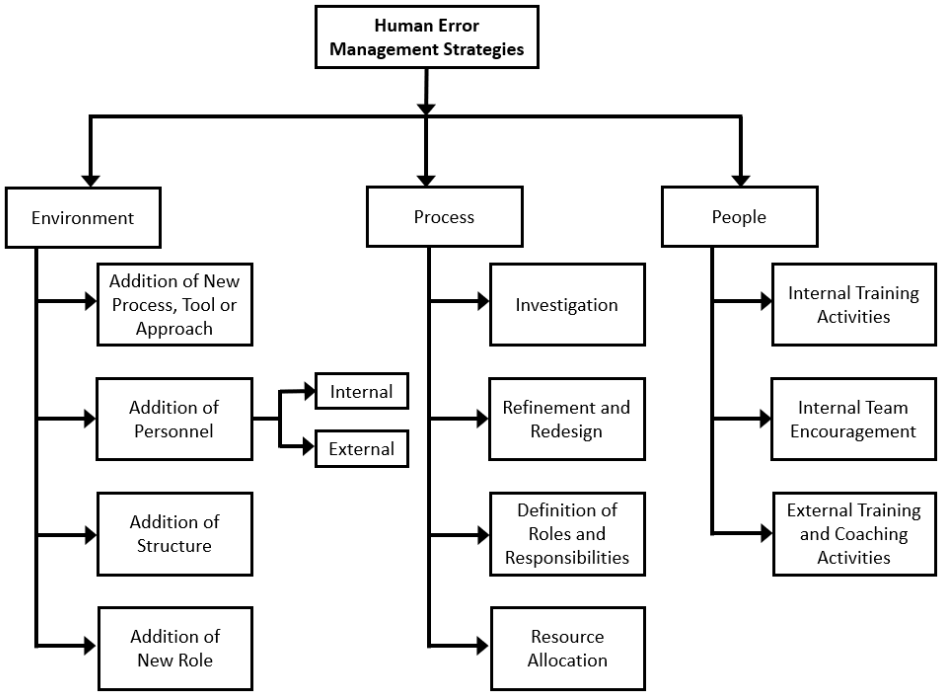


图 2：人为错误管理分类体系

分别占总数。八项策略未提供足够的信息，因此我们将其归类为模糊。表3中的列 **总计** 显示了属于各类别的策略总数。

步骤 *2*：将预防与缓解策略归入低级类别和需求阶段

我们进一步将每一类策略归入低层类别。低层类别表示为解决需求问题中的错误而必须进行的具体人员、流程和环境变更。最大的类别包含70项策略，即流程：细化与重新设计（*RR*）。该低层类别之所以如此庞大，其中一个原因是它包含了建议对沟通、文档记录、评审及其他流程进行变更的策略，正如第5节所述。第二大的类别是人员：内部培训活动（*ITA*），包含26项策略；其次是环境：新增流程、工具或方法（*ANP*），包含18项策略。尽管ITA比ANP包含更多策略，但我们发现总体而言，专注于人员的低层类别并不多。此外，大多数人员策略都与培训有关，

表 3：低层类别和需求活动中的预防与缓解策略

P/M Classes	P/M Sub-Categories	Requirement Activities					Total
		Elicitation	Analysis	Specification	Validation	Management	Unknown/ out of scope	
Environment	Addition of New Process, Tool or Approach	0	6	0	5	7	0	18
	Addition of Personnel (Internal)	0	2	0	0	0	0	2
	Addition of Personnel (External)	2	0	0	0	4	0	6
	Addition of Structure	1	1	4	2	0	1	9
	Addition of New Role	0	0	0	0	1	0	1
	Sub-total	**3**	**9**	**4**	**7**	**12**	**1**	**36**
Process	Investigation	0	1	0	0	6	0	7
	Refinement and Redesign	13	8	4	15	27	3	70
	Definition of Roles and Responsibilities	0	0	0	0	2	0	2
	Resource Allocation	0	0	0	0	0	0	2
	Sub-total	**15**	**9**	**4**	**15**	**35**	**3**	**81**
People	Internal Training Activities	0	0	0	0	26	0	26
	Internal Team Encouragement	0	2	0	0	2	0	4
	External Training and Coaching Activities	2	0	0	0	2	0	4
	Sub-total	**2**	**2**	**0**	**0**	**30**	**0**	**34**
Total		**20**	**20**	**8**	**22**	**77**	**4**	**151**

这可能是因为，要在流程或环境中实施变更，人员需要接受培训，以落实这些变更。因此，结果表明，与改变人员相比，着重改变流程和环境，对于有效预防和减轻需求工程中的人为错误更为重要。

接下来，为帮助需求工程师找到应实施变更的活动，我们将预防与缓解策略分配到需求工程生命周期中。以第1步的结果为起点，第一作者将预防与缓解策略分类到低层级类别中。她还将每项策略归类到受该变更影响的需求工程生命周期阶段。根据现有来源 [21, 22], 我们将需求工程生命周期的阶段定义为：

1.获取：包括需求收集、理解、组织以及与客户互动；2.分析：包括需求分类、确定优先级以及解决任何误解；3.规格说明：包括定义和记录用户需求与系统需求、实施需求的步骤，以及测试和验证实施情况的方法；4. 验证：包括对已规定需求的核查和验证，以检查其是否存在歧义、不一致和冗余；5. 管理：包括总体管理活动，例如管理计划、资源收集与分配以及变更管理；以及6. 未知：包括未明确说明相关需求活动的策略。

随后，另外两位作者审阅了该分类，并记录了所有不一致之处。与我们执行第1步的方式类似，我们将剩余的153项策略分成4批，分别包含15、30、49和59项策略。四批合计共有29处不一致，我们通过讨论会议解决了这些问题。在此阶段，我们删除了两项仍缺乏足够信息以进行分析的策略。这一变更使我们得以将151项策略映射到需求活动上。

在这151项策略中，我们将51%映射到管理 活动，15%映射到验证 活动，13%分别映射到获取 和分析 活动，

以及 5% 的规格说明 活动。只有 3%（或 4 个）策略属于未知 类别。

为总结研究结果，我们在在线附录[1]中提供了所报告策略的完整列表。对于每项策略，附录列出了相关的需求问题、其所针对的底层错误类型，以及它属于预防策略还是缓解策略。

5. 讨论

本节重点介绍研究结果中的关键观察发现。

5.1. 对策略进行分类

在分析这些策略时，我们注意到，由于提供的细节不足，而且有些策略之间的差异十分细微，因此分类并非总是易事。例如，改变环境 通常也需要改变流程，而要理解流程 的改变，往往也需要改变人员。不过，我们还是成功地将大多数策略归入单独的类别，只有少数策略被归入模糊 类别。

值得注意的是，第4节中分析的所有策略都要求引入或定义流程、角色或职责。基于先前发现移除子任务是应对错误的一种策略的研究 [14], ，我们预计也会看到要求移除流程步骤、角色或职责的策略。然而，我们的数据集中没有任何策略侧重于移除这些项目。虽然“细化”活动可能通过移除步骤来修改流程，或者“引入”活动也可能隐含地包括移除现有活动，但数据集中没有明确提及移除步骤。这一点值得进一步研究。

5.2. 研究启示

表3显示了针对各项需求活动所报告的策略数量。在总计159项策略中，8项属于模糊类别（未在表中显示），151项属于人员、流程和环境类别。类别），在这151项策略中，81项属于流程类别。换言之，从业者处理需求错误时，略多于一半的情况下会通过对流程实施变更来解决。在流程策略中

[1]http://carver.cs.ua.edu/Data/HumanError/Prevention-Mitigation-Strategies.html

类，“定期沟通活动/会议的规划”和“定期沟通活动/会议的执行”是最常见的错误管理机制，这表明大量需求问题可以通过更好的沟通来解决。流程类中下一个最常见的策略是“采用正式评审”，这表明需求验证和评审是避免需求工程错误的一项重要实践。此外，错误管理发生的第二常见需求工程活动是验证，这一事实也支持了这一推论。

通过将这些策略分配到各项需求工程活动中，我们的研究结果有助于从业者更好地识别每个阶段最适用的错误管理策略。如表 3 所示，77 项策略（数量最多的一组）适用于管理 活动，另有 22 项策略适用于验证 活-动。对于管理 活动，这些策略主要涉及流程和人员方面的变更。然而，对于验证 活动，没有与人员变更相关的策略。表 3 显示，在管理 活动中采用的 77 项策略里，只有 30 项侧重于人员变更。此外，在其他需求活动中，采用的与人员相关的策略要么没有，要么很少。与人员相关的策略集中在管理活动中，这只是因为许多管理活动（例如组织培训和会议）都与人员变更有关。这些观察结果表明，在各项需求活动中，需求工程师主要通过与流程 和环境 变更相关的策略来管理人为错误。将这一观察结果与第 2 节所述医学领域的文献联系起来，我们的研究结果支持这样的观点：改变人员并不是管理人为错误的有效方法 [9]。

接下来三个最常发生错误管理的需求工程活动是获取、分析 和规格说明。获取过程中采用的大多数策略都属于低层级类别流程：细化与重新设计。分析和规格说明活动所采用的策略则同样侧重于改变流程 和环境。这些策略的分布再次表明，从业者更关注改变流程 和环境，而非改变人员。

表3显示错误管理中最常见的变更类型是过程：细化与重新设计 *(RR)*。由于该类别包含大量策略，我们试图进一步划分它。然而，由于数据有限且模糊，我们无法进一步分解它。专业需求工程师大多应用RR类中的策略这一事实表明

他们认为这些策略有效且成功。

除 RR 外，接下来最常见的一组策略是人员：内部培训活动（*ITA*）。这一结果表明，从业者认为，通过培训提升技能，是管理大多数与人员 相关错误的最佳方式。人员 类别中的另外两个低层类别，仅包含少量预防与缓解策略。

接下来最常见的变更类型是环境：新增流程、工具或方法（*ANP*）。为任务提供结构，是环境 类别中另一种有效的错误管理机制。这里的关键发现是，尽管许多需求错误可以通过修复现有流程来解决（如上文所述），但有时仍需要制定新流程。引入新流程通常需要调整现有流程，并需要培训人员以适应这些变化。

我们还观察到，如表 3 所示，环境 类别中的预防与缓解策略较为均匀地分布在各个低层类别中。然而，在人员 类别中，这些策略仅出现在少数几个低层类别中，且大多集中于人员：内部培训活动（*ITA*）。这种分布表明，为了管理人为错误，从业者应关注其他重要类型的变更，例如环境和流程方面的变更，而非人员方面的变更。根据其他领域的文献（第 2 节），这一启示是积极的 [8, 9, 10]。一项探讨如何管理人为错误的医学领域研究发现，以改变人员为重点的解决方案在处理错误方面鲜少奏效 [9]。该研究还指出，着眼于技术、工具和政策变更的错误管理方法更为有效，但较少采用 [9]。我们的分析和观察也得出了类似启示。因此，我们的结果回答了标题中的问题我们应该改进人员、流程还是环境？，建议优先改进流程。

5.3. 观察到的特征与维度

我们观察到，RR 子类别在流程中表现出若干特征，包括沟通、文档记录、规划与组织，以及审查和交叉核对。这一观察意味着，大多数时候，需求工程师可以通过重新设计或重新定义沟通、文档记录、规划或审查流程，来克服需求错误和问题。在分析这些策略时，我们尝试将较大的 RR 类别划分为更小的类别。然而，我们发现，可以根据这些策略识别出更多更详细且彼此 distinct 的类别。例如，“使用检查清单

来监控需求的整个生命周期”可以理解为：通过维护检查清单体现“文档记录”特征，同时也通过监控需求体现“审查和交叉核对”特征。因此，我们没有进一步分解 RR 类别，因为数据集中的信息未能提供足够的细节来进行进一步分析。

在我们的分析中，我们还确定，可以从谁、做什么和如何的视角来看待预防与缓解策略。谁视角关注由谁发起变更，以及由谁执行变更。做什么 视角关注变更本身。如何视角关注如何实施变更。我们预计，这些视角将有助于从业者理解谁应负责实施必要的变- 更。由于我们样本中的数据不够详细，无法清晰识别这些视角，因此我们未能将这一维- 度纳入我们的工作。不过，我们确实认为，在未来的工作中探索这一方向将很有价值。

5.4. RET、*HET* 和 *HEMT* 的比较

本小节介绍三种分类体系的概览——需求错误分类体系（*RET*）、人为错误分类法（*HET*）和人为错误管理分类体系（*HEMT*）。RET 为将错误作为软件故障根本原因的研究提供了起点 [18]。HET 从人的认知视角对 RET 中的研究进行了拓展，以调查并分类错误 [7]。在当前研究中，HEMT 延续了 HET 的研究，引入错误管理分类体系，并支持对 HET 中错误类型的错误预防和缓解。表 4 展示了上述分类体系之间的比较。

6. 效度威胁

本节描述了本研究的效度威胁以及我们如何应对这些威胁。

6.1. 内部效度

调查数据是主观的。我们不知道所报告的策略是否有用，也不知道受访者对其答案的信心如何。然而，由于参与者是来自不同软件行业和不同国家的经验丰富的工业开发人员，我们认为这一威胁是最小的

表 4：RET、HET 和 HEMT 的比较

Aspect	RET	HET	HEMT
Focus	Types of requirements errors that affect overall software quality.	Human errors based on classification system from Cognitive Psychology.	Human error prevention and mitigation approaches in requirements engineering based on HET.
Approach	Bottom-up approach without a guiding error theory.	Extension of the RET People class with detailed categorization of human errors in requirements engineering	Complements HET by providing formal taxonomy of human error management classes and subcategories
Organization	Organizes requirements errors into People, Process, and Documentation categroies.	Organizes human errors in requirements engineering into Slips, Lapses, and Mistakes [1].	Organizes human error management strategies into changes to People, Processes, and Environment.
Construction Process	Systematic literature review of software engineering research.	Systematic Literature review of software engineering and cognitive psychology research.	Analysis of survey data from practitioners reporting actual experiences by the practitioners.
Use	Supports fault detection, error detection, and error prevention for requirements errors. Because these errors are not true human errors, errors can still occur resulting in faults.	Supports fault detection, error detection, and error prevention. The human error information in the HET will help requirement engineers avoid making such errors and hence prevent the faults caused by such errors.	Supports error prevention and error mitigation. It provides requirements engineers with knowledge about strategies to prevent and mitigate different types of errors.

相对于本文的目标。此外，我们丢弃了任何包含模糊解释或与我们的研究问题无关的解释的回复。为了防止潜在的确认偏误，即只包含那些确认我们目标的回复，三位作者分别分析了数据并就我们剔除的项目达成一致。然而，我们确实需要进一步的研究来评估所报告策略的有用性。

6.2. 构念效度

对于NaPiRE数据集，受访者提供了他们如何预防和缓解需求问题的信息。然而，该调查并未专门关注人为错误。因此，在早期工作中，两位作者将这些问题映射到人为错误 [4]。由于我们没有直接与受访者互动，先前的工作或我们当前的工作可能误解了调查中提供的信息。此外，一些回复不是英文的，另一些则是模糊的。为了最小化这一威胁，我们只考虑英文回复，并为模糊回复创建了一个单独的类别。

在CAPS调查中，参与者首先接受了有关需求工程中人为错误的培训，然后回答有关预防和缓解策略的调查问题。即便如此，他们仍可能未能充分理解人为错误，因此提供了不相关的信息。

为应对这些威胁，我们结合调查中有关问题原因和错误类型的其他信息，对受访者报告的策略进行了分析，以便正确解读他们的回答。三位作者分别独立进行了分析，随后共同讨论并解决了任何分歧。这些不同的视角也降低了这一威胁带来的潜在影响。

6.3. 外部效度

由于设计特点，NaPiRE调查提供的信息涵盖了来自世界各地的软件组织，具有广泛的代表性。这种多样性提高了所报告结果具有外部效度的可能性。相反，CAPS调查的数据仅代表一个小型组织的信息。本文使用了两项调查的数据得出结论。由于本研究的主要目标是基于工业实践者的数据提出一个分类体系，因此外部效度方面的威胁较低。此外，该分类体系基于从业者用于管理实际需求问题的真实错误管理策略构建。因此，外部效度方面的威胁

较低，尤其是分类体系能否反映真实策略这一点。不过，我们也指出，构建该分类体系只是更大规模研究计划的第一步，后续还需要开展更多研究来验证（并改进）这一分类体系。

7. 结论

由于需求工程过程中发生的人为错误往往要到很久之后才会表现为故障，并且可能一直隐藏到执行阶段，因此很难识别这些错误。从业者需要尽早在软件开发过程中处理人为错误，以降低成本并提高软件质量。需求工程中关于人为错误的研究不足，使需求工程师难以了解这些错误的存在以及管理它们的技术。本研究对源自工业实践的错误管理机制进行了研究，并初步构建了错误管理方法的分类体系。

我们的分析结果归纳出预防与缓解策略的三个高层级类别：人员、流程和环境。在每个高层级类别中，我们还识别出一组更低层级的类别，以进一步细分这些策略。由此形成的分类体系，为需求工程中的人为错误管理方法提供了一定的结构。未来的研究可以此结构为起点，评估各项预防与缓解策略的有效性，并找出分类体系中缺失的类别。此外，了解错误管理方法在各项需求活动中的分布情况，将有助于研究人员和从业者更好地理解哪些方法最适用于哪些活动。

该研究还深入揭示了行业中所采用的错误管理方法。尽管人为错误被定义为人类认知中的错误，通常与人员相关，但调查中确定的大多数错误管理策略都侧重于改变流程，而非改变人员。鉴于流程类别中的策略占比较高，且这些策略涵盖了大多数类型的人为错误，我们可以推断，与其他类别的策略相比，与流程相关的错误管理策略可能更为有用。这一观察结果与医学领域的研究一致，该研究发现，人员相关解决方案不如流程相关解决方案成功 [9]。

本文介绍了一项关于需求工程中人为错误管理的初步研究，具体而言，也涉及软件工程领域的相关议题，

更广泛地说。我们根据从业者描述的需求问题及其解决方案构建了该分类体系。因此，该分类体系本身具有价值，既能系统整理这些信息，也能作为进一步探索软件工程中人为错误管理的研究基础。分类体系中的所有条目，包括每个低层级类别，都需要通过研究加以考察，而这些研究应围绕预防与缓解策略设定更具体的目标。

在未来的工作中，我们计划在工业环境中评估该分类体系的有效性、实用性和完整性。为进行这项评估，我们打算让从业者使用该分类体系，帮助他们预防或缓解项目中遇到的问题。随后，我们将调查该分类体系中的错误管理策略在多大程度上帮助从业者克服他们面临的需求问题。我们预计，通过实证研究，可以比较各类预防与缓解策略的有效性，从而找出最有效的策略，为从业者提供指导。

参考文献

[1] J.Reason，《人为错误》，剑桥大学出版社，1990。[2] S. Viller，J. Bowers，T. Rodden，需求工程中的人为因素，技术报告，技术报告，协作系统工程组，CSEG/8/1997（1997）。[3] Z.Askarinejadamiri，H. Zulzallil，A.A. A. Ghani，K. T. Wei，人为错误对软件需求波动性的影响，《国际先进计算机科学与应用杂志》8（2）（2017）227–237。[4] W. Hu，J. C. Carver，V. Anu，G. S. Walia，G. L.Bradshaw，利用人为错误信息进行错误预防，《实证软件工程》23（6）（2018）3768–3800。[5]D. M. Fern´andez，S. Wagner，M.Kalinowski，M.Felderer，P.Mafra，A. Vetr`o，T. Conte，M.-T. Christiansson，D. Greer，C. Lassenius，等，指出需求工程中的痛点，《实证软件工程》22（5）（2017）2298–2338。[6] B. Kitchenham，D. I. Sjøberg，O.P.Brereton，D.Budgen，T.Dyb˚a，M. H¨ost，D. Pfahl，P.Runeson，我们能否评估软件工程实验的质量？，载于：2010年ACM-IEEE实证软件工程与测量国际研讨会论文集，2010年，第1–8页。

[7] V. Anu, W. Hu, J. C. Carver, G. S. Walia, G. Bradshaw，软件需求人为错误分类体系的开发：系统文献综述，Information and Software Technology 103 (2018) 112–124.[8] J. Gosbee, T. Anderson，人因工程设计演示可以启发你的根本原因分析团队，BMJ Quality & Safety 12 (2) (2003) 119–121.[9] K. M. Kellogg, Z. Hettinger, M. Shah, R. L. Wears, C. R. Sellers, M. Squires, R. J. Fairbanks，我们当前的根本原因分析方法是否导致我们未能改善患者安全？，BMJ quality & safety 26 (5) (2017) 381–387.[10] A. W. Wu, A. K. Lipshutz, P. J. Pronovost，医学领域根本原因分析的有效性和效率，Jama 299 (6) (2008) 685–687.[11] H. W. Hendrick，通过控制器、显示器和工作空间布局的人体工程学设计减少人为错误，载于：印度尼西亚生理学专家协会第十一届全国大会暨第十三届学术研讨会及人体工程学与运动生理学国际研讨会论文集，2002，第 14–17 页.[12] F. Huang, B. Liu, B. Huang，用于识别软件缺陷人为错误成因的分类系统，载于：第 18 届国际可靠性与质量设计会议，2012，第 44–49 页.[13] F. Huang, L. Bin，基于人为错误理论的软件缺陷预防，Chinese Journal of Aeronautics 30 (3) (2017) 1054–1070.[14]F. Huang，软件开发中的完成后错误，载于：2016 IEEE/ACM 软件工程合作与人为因素会议（CHASE），IEEE，2016，第 108–113 页.[15] J. Back, W. L. Cheng, R. Dann, P. Curzon, A. Blandford，避免程序错误的动机是否会影响其系统性？，载于：人员与计算机 XX——参与，Springer，2007，第 151–157 页.[16] D. Firesmith，常见需求问题、其负面后果以及有助于解决这些问题的行业最佳实践，《Journal of Object Technology》6 (1) (2007) 17–33.[17] M. E. R. F. Lopes, C. H. Q. Forster，人为错误理论在需求工程流程改进中的应用，Information Sciences 250 (2013) 142–161.

[18] G. S. Walia, J. C. Carver, 一项识别和分类软件需求错误的系统文献综述, Information and Software Technology 51 (7) (2009) 1087–1109.
[19] F. Lanubile, F. Shull, V. R. Basili, 在需求文档中实验错误抽象, in: Proceedings 第五届国际软件度量研讨会. Metrics (Cat. No. 98TB100262), IEEE, 1998, pp. 114–121.[20] G. S. Walia, J. C. Carver, 评估使用需求错误抽象与分类方法以在制品创建期间防止错误：一项可行性研究, in: 2010 IEEE 第21届国际软件可靠性工程研讨会, IEEE, 2010, pp. 81–90.[21] G. Kotonya, I. Sommerville, 需求工程：流程与技术, Wiley Publishing, 1998.[22] P. Parviainen, H. Hulkko, J. K¨a¨ari¨ainen, J. Takalo, M. Tihinen, 需求工程：技术清单, VTT Technical Research Centre of Finland, 2003.

附录 A. NaPiRE 数据集中报告的需求问题

问题编号	评论需求问题
1	项目团队内部的沟通缺陷
2	我们与客户之间的沟通缺陷
3	术语问题
4	职责不明确
5	不完整和/或隐藏的需求
6	项目负责人提供的支持不足
7	客户支持不足
8	难以将需求与 此前已知的解决方案设计
9	需求不一致
10	缺少可追溯性
11	目标不断变化（目标、业务流程和／或需- 求）
12	“镀金”（实现了没有相应要- 求的功能）
13	无法充分获取客户需求和/或（内部）业务信息, 化
14	对客户应用领域了解不足
15	与客户的关系薄弱
16	时间盒管理 / 总体上时间不足
17	高度创新程度与对.....的需求之间的差异- 不当接受（可能错误 / 不完整 / 未知） 需求
18	技术上不可行的需求
19	规定不充分的需求过于抽象，并且允许 多种解释
20	不明确或不可测量的非功能需求
21	关于客户业务领域的不稳定因素，例如变化中的 联系点、业务流程或需求