



计算机科学

COMPUTER SCIENCE

软件需求工程技术综述

王浩宇

引用本文

王浩宇. 软件需求工程技术综述[J]. 计算机科学, 2022, 49(11A): 210900132-14.

WANG Hao-yu. Review on Technologies of Requirement Engineering of Software[J]. Computer Science, 2022, 49(11A): 210900132-14.

相似文献推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[结合情感信息的个性化对话生成](#)

Personalized Dialogue Generation Integrating Sentimental Information

计算机科学, 2022, 49(11A): 211100019-6. <https://doi.org/10.11896/jsjcx.211100019>

[预训练语言模型的扩展模型研究综述](#)

Survey of Research on Extended Models of Pre-trained Language Models

计算机科学, 2022, 49(11A): 210800125-12. <https://doi.org/10.11896/jsjcx.210800125>

[基于双图神经网络信息融合的文本分类方法](#)

Text Classification Method Based on Information Fusion of Dual-graph Neural Network

计算机科学, 2022, 49(8): 230-236. <https://doi.org/10.11896/jsjcx.210600042>

[中文预训练模型研究进展](#)

Advances in Chinese Pre-training Models

计算机科学, 2022, 49(7): 148-163. <https://doi.org/10.11896/jsjcx.211200018>

[自然语言处理在简历分析中的应用研究综述](#)

Survey of the Application of Natural Language Processing for Resume Analysis

计算机科学, 2022, 49(6A): 66-73. <https://doi.org/10.11896/jsjcx.210600134>

软件需求工程技术综述

王浩宇

武汉数字工程研究所 武汉 430000

摘要 需求工程作为软件项目的首个过程,它的实施质量很大程度上能决定软件项目的成败。80年代中期敏捷软件开发的出现使得需求工程从由计划驱动过渡到现在的由价值驱动。此外,硬件性能的大幅提升与人工智能的再度兴起使自然语言处理的效率和规模与日俱增,需求工程更能本地易地利用自然语言处理来分析大量文本数据。物联网、边缘计算和大数据的出现使投资者和开发人员更容易获取到大量用户数据和商业信息,数据驱动的需求工程作为一种新概念也逐渐被业界所知。文中首先综述了需求工程的发展历程,包括从需求工程方法学开始,到面向对象建模,再到基于本体和面相特征的需求工程,以及与机器学习相关的自动化需求提取技术。随后重点介绍了3种需求工程相关的研究方向,包括需求工程中采用的自然语言处理方式和部分支持的语言类型,敏捷需求工程的发展历程以及近年提出的工具和方法,数据驱动的需求工程的概念、必要性、过程和近年主要采取的方法和实践。最后基于一些各国关于需求工程开展情况的报告,分析了需求工程近年遇到的困难和挑战,并对需求工程未来的发展进行展望。

关键词:需求工程;敏捷开发;自然语言处理;数据驱动

中图法分类号 TP311.5

Review on Technologies of Requirement Engineering of Software

WANG Hao-yu

Wuhan Digital Engineering Research Institute, Wuhan 430000, China

Abstract As the first process of software project, requirements engineering's implementation quality can determine whether a software project can success or not. Requirements engineering was put forward in the 1980s the first time, and the model used also transited from the earliest process oriented model to the object-oriented model widely used in industry, and then to the service-oriented model proposed and gradually promoted after 2004. In addition, with the rapid improvement of hardware performance and the resurgence of artificial intelligence, the efficiency and scale of natural language processing are increasing, which makes it easier for requirements engineering to use natural language processing to analyze large amounts of text data. The emergence of IoT, edge computing and big data makes it easier for investors and developers to obtain a large amount of user data and business information. As a new concept, data-driven requirements engineering is gradually known by the industry. This paper reviews the development history of requirements engineering at first, including requirements engineering methodology, object-oriented modeling, requirements engineering based on ontology and facial features, and automatic requirements extraction technology related to machine learning. Then it focuses on three research directions of requirements engineering, including the natural language processing methods and some supported language types, the development history, tools and methods proposed in recent years of agile requirements engineering, as well as the concept, necessity and process, the main methods and practices of data-driven requirements engineering in recent years. Finally, based on the reports on the development situations of requirements engineering in some countries, this paper analyzes the difficulties and challenges of requirements engineering in recent years, and prospects the future development of requirements engineering.

Keywords Requirements engineering, Agile software development, Natural language process, Data-driven

1 引言

需求工程是系统工程中的重要阶段。良好的流程是需求工程的关键,它决定了产品生成的效率和速度。

需求是每个项目的基础,它定义了潜在新系统中的利益相关者(用户、客户、供应商、开发人员、企业)需要从系统中获得什么,以及系统必须完成什么操作才能满足这种需求。为了让每个人都能很好地理解,它们通常使用自然语言来表达^[1]。

需求工程作为软件产品线领域工程的第一环也是最重要

的一环,其在软件开发过程中的重要性不言而喻。根据美国专门从事IT项目跟踪的权威机构Standish Group在1995年和1996年的调查报告显示:有51.6%的IT项目失败与需求直接相关,而有42.4%的IT项目成功与需求直接相关。其中需求调研不充分,用户需求描述不完整、不准确,不能跟踪需求是否被正确地实现,或者不能很好地管理需求变更都是软件开发失败的主要原因^[2]。进入20世纪90年代后,需求工程已成为软件界研究的热点之一。从1993年起每两年举办一次需求工程国际研讨会,自1994年起每两年举办一次

需求工程国际会议(ICRE),在1996年Springer发行了需求工程权威书籍“Requirements Engineering”。本文第2节介绍需求工程整体的发展历程,包括学术界的研究和工业界的应用现状;第3节介绍3种需求工程的主流研究方向及其研究现状,包括对自然语言建模的需求工程、敏捷需求工程(ASD)和较为新颖的数据驱动的需求工程;第4节介绍需求工程目前面临的困难与挑战,并对需求工程未来发展做出了展望。

2 需求工程的发展历程

随着信息时代相关企业业务需求的发展,信息种类和数量高速增长,用户的需求也日趋复杂,需求变更成为必然,原有的注重实现的模式已经不能适应现代软件工程的需要。

由于学术界和工业界的重视,需求工程发展势头迅猛,在其方法学方面有一些典型的研究成果^[2]。Lano^[3]提出的操作概念规格,于需求产生前由开发人员完成,它既满足精确的规格说明要求,同时易读、易理解,便于用户了解是否真正体现了其要求。Sutcliffe^[4]、Maiden等^[5]提出从领域知识的角度定义在需求工程环境中通用的领域语义模型和组合模型。Alford^[6]提出任务分割的概念,大大降低了需求分析的问题复杂度。Chou等^[7]和Eckert^[8]讨论了面向对象的需求工程方法学的概念和模型。Drake等^[9]提出用于确定系统需求边界的限定过程。Gotel^[10]对需求跟踪性问题进行了研究。

面向对象的方法是需求分析中的一个热点。Yourdan等的OOA方法、Booch的方法、Jacobson的OOSE、Rumbaugh的OMT方法等,都是这一方法的典型。Coad和Yourdon采用5个步骤来确定一个多层的OO模型,5个步骤分别对应模型的5个层次。它是最早的面向对象的分析与设计方法之一,该方法适合面向对象的初学者使用,但由于该方法在处理能力方面的局限,目前已经很少使用。Booch是面向方法最早的倡导者之一,他提出了面向对象的软件工程的概念。1991年,他将面向Ada的工作扩展到整个面向对象的设计领域,Booch的方法比较适用于系统的设计和构造。Jacobson于1994年提出了OOSE的方法,其最大特点是面向用例(Use-Case),并在Use Case的描述中引入了外部角色的概念。Use Case成为分析模型的基础,用交互图对Use Case进一步描述后就形成设计的模型。Rumbaugh等提出了面向对象的建模技术(OMT),采用面向对象的概念,引入各种独立于语言的表示符。这种方法用对象模型、动态模型、功能模型和实例模型共同完成对系统的建模。开发人员无须在开发过程的不同阶段进行概念和符号的转换。

目前基于本体和面向特征的需求工程方法得到广泛的重视。Lu等^[11]提出了一种面向本体的需求分析方法,该方法使用本体作为对象关联的手段,以增强面向对象方法的表达能力。Jin^[12]提出了一种基于本体的需求获取方法。目前也有很多面向特征的需求分析方法,Kang等^[13]提出了一种面向特征的领域分析方法FODA。Gomaa等^[14]介绍了一种领域建模方法及相应的建模环境,在基于特征的可变性建模方法中,主要关注特征模型的组织结构、特征之间的关系、特征与核心资产的关联等。各种方法主要扩展了基于特征的建模方法,在特征之间的关系表示和特征与核心资产的关联方面开展研究。其中,Robak等^[15]、Papiaewskih等^[16]、Czarneckil等^[17]和Von Der Malen等^[18]主要从特征之间的关系角度

进行了研究。Mezini等^[19]和Hartmann等^[20]对于复杂系统的建模支持进行了研究。Czamecki等^[21]、Hammouda等^[22]、Lee等^[23]和Dhuangana等^[24]主要从特征与核心资产的关联角度进行了研究。大多数研究的重点在于特征建模、特征分析和特征依赖的建模和分析等。

21世纪后,通信技术和计算机算力迅猛发展,随着边缘计算、大数据、物联网等新技术的提出,以及人工智能热度的回升,需求分析也被这些新技术赋予了新的生命力。Dag等^[25]提出一种通过分析大量用户群体反馈来改进需求的方法。Eduard等^[26]在此基础上提出了针对大量用户群体反馈的需求工程理论,它寻求一种能够收集Web2.0时代的新兴媒体、物联网嵌入式系统领域以及智能领域的大量用户反馈,并加以分析以改进服务提供者服务的方法。Andreas^[27]提出了一种用于自动化和云项目的需求分析方法,它较之前类似的方法更能考虑到服务提供商的需求并能分别实现。Ferrari等^[28]尝试用机器学习和自然语言处理方法来解决日趋复杂的需求工程任务,Jonas等^[29]对其加以改进,以使机器学习处理的精度和召回率达到较好的平衡。此外,分类器也常常被用于需求工程,在过去的几十年里,分类器被设计和使用以达到各种目的,包括检测和分类非功能需求性(NFRs)^[30-33],分析应用程序评论以区分bug、特征和表扬^[34]或识别有用的评论^[35],以及将需求与大型规范文档中的其他信息分离。

需求工程的目的是通过与用户广泛地交流来确定应用系统的目标。需求活动以“工程化”的方法被提出、分析和组织,它鼓励用户以一种积极的方式参与到需求分析活动中,并在整个软件生命周期强调用户参与和领域专家的指导作用,促使目标系统最大化地满足用户需求。

3 需求工程研究方向及研究现状

本节将从需求工程的3个研究方向,包括自然语言处理与需求工程、敏捷需求工程(ASD)和数据驱动的需求工程,分别介绍他们的特点、研究现状和应用情况。

3.1 自然语言处理与需求工程

软件需求是对在给定的软件系统中设计和开发的必要功能的描述^[36]。如果需求中包含模糊声明,则项目的利益相关者无法清楚地理解它,客户和软件团队之间可能会出现交流问题^[37],这些问题还可能导致软件开发生命周期(SDLC)下一阶段的延迟,并增加软件项目的成本。因此,必须将这些需求转化为易于理解的概念模型^[38]。通常,系统分析人员在定义所有用户需求后,手动设计软件系统的概念模型。它们努力提取具有属性和关系的实体。近年来,自动将软件需求转化为概念模型的研究显著增加^[39-41]。

随着互联网资源变得更加容易获取,设计有效的跨语言信息的系统方法也是研究重点之一^[42]。在需求分析中提供使用多种语言的通用方法,可以实现广泛的覆盖和可访问性。然而目前大多数工具只能用于分析英语表示的需求,因为它是世界上使用最广泛的语言。此外,与其他语言相比,英语的表达形式更简单,所以更容易获得比其他语言更准确的分析结果。也有一些研究支持其他语言,如文献^[43,46-48]分析了土耳其语的需求,文献^[46]分析了西班牙语需求,文献^[47]分析了德语需求,文献^[48]分析了泰语需求。考虑到当前

研究中的语言局限性,文献[49]建议创建一个共同的规则集,支持分析用任何语言编写的需求,以促进跨语言实践。该文献中还指出一些转换规则可以独立于语言在概念识别研究中加入以应用。

在将需求转化为概念模型方面,大部分研究普遍采用类似的方法。这些方法包括 NLP(自然语言处理)分析模型和基于规则的模型,如图 1 所示。

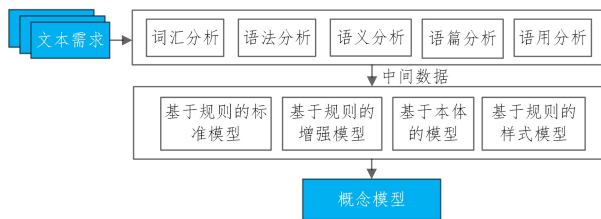


图 1 使用自然语言处理方法分析文本需求的一般流程

Fig. 1 Common procedure of analyse text requirements with NLP

首先,使用 NLP 分析模型预处理需求,以获得中间数据。然后,将中间数据作为基于规则的转换模型的输入,最终自动生成概念模型。有 5 种 NLP 分析模型可用于预处理文本

需求:词汇分析、语法分析、语义分析、语篇分析和语用分析^[50]。文献[40]介绍了一种利用词汇分析、语法分析和语义分析的 NLP 模型,结合基于规则的增强转换模型,将英语文本转换为类的方法。文献[44]介绍了一种利用相同 NLP 模型,结合基于规则的标准转换模型,将英文文本转换为用例(Use-case)的方法。文献[46]介绍了一种利用语法分析和语义分析 NLP 模型,结合基于规则的标准转换模型,将以西班牙语表达的用例转换为类的方法。文献[47]介绍了一种利用词汇分析 NLP 模型,结合基于规则的增强转换模型,将德语文本转换为类和用例的方法。

概念模型是描述软件设计元素的一种替代方案。它为正在运行中的所提议的模型提供指导。该模型可以有多种形式,如 UML 图、实体关系模型(ERM)和业务模型(BM)。包括 14 种图类型的 UML 是用于指定软件模型的结构和行为的最常用的概念模型。大部分研究生成的概念模型通常以 UML 类图的形式存在。文献[47,51-53]在生成 UML 类图之外还生成其他类型的 UML 图。文献[51,54-57]除了生成 UML 类图外还支持代码生成。此外,表 1 列出了近年来相关研究所使用方法的信息。

表 1 相关研究方法和输出

Table 1 Related research methods and outputs

文献	发表时间	需求	NLP 模型	转换模型	输出
[37]	2014	文本(英语)	LexA, SynA, SemA	增强,基于规则	类
[39]	2018	文本(英语)	LexA	样式,基于规则-	类
[40]	2019	文本(土耳其语)	LexA, SynA	本体,样式,增强,基于规则	类
[41]	2020	文本(英语)	LexA, SynA, SemA	本体,基于规则	用例
[42]	2018	文本(英语)	LexA, SynA	标准,基于规则	用例
[43]	2018	文本(英语)	LexA, SynA	样式,基于规则	领域模型
[44]	2019	文本(英语)	None	样式,增强,基于规则	活动
[46]	2008	用例(西班牙语)	SynA, SemA	标准,基于规则	类,用例
[47]	2004	文本(德语)	LexA	增强,基于规则	类
[48]	2017	文本(泰语)	LexA	标准,基于规则	对象
[49]	1996	文本(英语)	LexA, SynA, SemA	标准,基于规则	类
[50]	2001	文本(英语)	LexA, SynA	标准,基于规则	类
[51]	2000	文本(英语)	None	样式,基于规则	对象
[52]	2002	文本(英语)	None	样式,基于规则	类,序列,Java, C++
[55]	2002	用例(英语)	None	样式,基于规则	类,序列
[56]	2002	文本(英语)	SemA, DiscourseA	样式,基于规则	类
[57]	2003	文本(英语)	LexA, SynA, SemA	本体,基于规则	类
[58]	2010	用例(英语)	LexA	增强,基于规则	活动
[59]	2004	文本(英语)	LexA, SynA, SemA	增强,基于规则	类,序列
[60]	2004	文本(英语)	LexA, SemA	样式,基于规则	类

注:LexA 为词汇分析, SynA 为语法分析, SemA 为语义分析, DiscourseA 为语篇分析

3.2 敏捷需求工程

80 年代中期,Takeuchi 等^[65]指出,产品开发的顺序阶段方法由于缺乏灵活性,已经不适合现代软件开发。此后,新的开发过程模型不断出现。例如迭代过程模型,如 Rational Unified Process^[66-67];此外还有一些敏捷方法,如 Scrum^[68-69]、极限编程(XP)^[70]、特征驱动的开发^[71]和 Kanban(看板)。2004-2010 年间,Kanban 在 IT 方面的使用主要受到 Anderson 的影响^[72-73]。近年来已经出现一些关于轻量级过程模型的研究。2001 年,这些不同流派的倡导者联合起来,创立了敏捷软件开发宣言^[67]。敏捷宣言包括有助于优化软件开发过程的价值观和原则,也对敏捷软件开发(ASD)中的团队协作产生了很大的影响^[74]。敏捷宣言提供了以下 4 个核心价值观:个人与交互重于流程和工具、有效的软件重于完整的文档、客户合作重于合同谈判、响应变化重于遵循计划。

自从敏捷开发兴起后,软件开发开始从计划驱动转向价值驱动,其过程模型如图 2 所示。

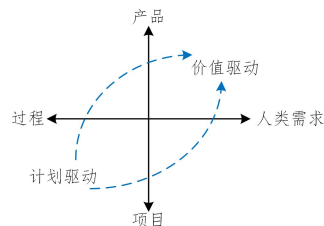


图 2 价值驱动模型与计划驱动模型在关注点上的区别

Fig. 2 Differences in focus between the value-driven model and the plan-driven model

在计划驱动的环境中,人们经常就模型定价、项目计划以及他们可以用现有资源开发多少产品特性进行谈判。它们强调产出(在一段时间内开发的特性的数量或发布的数量)。

而在价值驱动的环境中,人们关注的是前景、经验和人类价值观,以及他们如何通过产品来解决这些问题。它侧重于结果,这意味着它们侧重于产出所带来的差异。因此,敏捷开发主要是由人类的价值观所驱动。

在敏捷开发的背景下,需求工程是在整个开发过程中迭代地执行的,而不仅是拘泥于开发过程的开始阶段。为此需要一个实时模型用于将高级需求细化为开发人员可以实现的低级任务,使业务人员、利益相关者、用户和开发人员能以某种方式协同工作。该模型基于构件,从通过史诗的方法捕获需求开始。史诗相当于一个大型用户故事(User Story)^[76],可以利用故事图^[77]进行改进。故事图由用户故事或角色故事^[78]组成,它们被分为多个子任务。整个工作流可以使用kanban的方法设计、开发和交付^[79]。

如连续交付(CD)^[80]这样的敏捷开发技术会对现在的可用性测试的方式和频率产生影响^[81]。用户和系统本身在系统开发过程中提供新的信息,这些信息在接下来的步骤中被处理并为决策过程提供基础。因此,需求被视作假设,并被不断验证。还有一些研究侧重于RE和测试的一致性,这些研究调查了将测试用例作为需求的做法。在这种情况下,详细的需求通常被记录为测试用例,而不是使用额外的需求规范,以减少更新独立构件的工作量^[82]。

敏捷软件开发常用来与“以用户为中心设计(UCD)”集成,旨在提供更好的用户体验,在这个过程中,利益相关者和用户的参与是一个系统成功的关键因素^[83]。而且,与传统方法相比,这种参与不仅限于开发的早期阶段,因为利益相关者和用户参与了整个开发过程^[84]。对于敏捷需求来说,相关的研究不仅是技术层面的,也是社会层面的,大量研究致力于加强利益相关者和用户在软件开发过程中的参与。例如Bellucci等^[85]将XP与协同设计会话相结合,以开发具有强大用户参与能力的产品。他们在一项实地研究中探讨了用户如何与原型交互和工作,后来他们发现原型是迭代开发的。这种方法可以看作是一种快速部署进化原型的策略。Harbers等^[86]研究了价值故事研讨会在用户故事引出过程中的应用。价值故事研讨会的目的是将利益相关者价值嵌入到需求工程过程中。因此,必须先确定直接和间接的利益相关者。然后发掘每个利益相关者群体的价值数据。最后根据每个数据的情况分析利益相关者需求。Olsson等^[87]从多个案例研究中推导出了一个概念模型,该模型强调需要将开发早期阶段的定性反馈与开发后期阶段的定量客户观察相结合。在定性/定量客户驱动开发(QCD)模型中,需求被视为需要由连续的客户反馈所验证的假设。Dragicevic等^[88]声称商业用户可以收集必要的数据和文件(如模板、文档或屏幕的扫描件),并向以UML为这些数据建模的开发人员提供信息。协作和共同理解对ASD至关重要^[74]。Ramesh等^[90]开展的一项多案例研究显示非正式的和频繁的沟通是敏捷需求工程的核心。他们表示客户直接参与了每一次迭代。通过与客户面对面的沟通,提出、完善和验证了需求。他们还声称,需求分析是一个社会政治过程,它依赖于人类的互动,且受到几个语境因素的影响。

本文分析了2010年到2019年中相关研究论文共计27篇,他们所采用的研究方法如表2所列。

表2 论文方法统计

Table 2 References method statistics

研究方法	论文数量	占比/%
案例研究	15	56
实验室案例研究	2	7
多案例研究	2	7
方法研究	5	19
实验	1	4
实验视角	1	4
半结构化采访	1	4

敏捷需求工程中,收集用户反馈是很重要的一环,相关研究在收集用户反馈数据方面也使用了不同的方法与技巧。例如Bellucci等^[85]将每周的协同设计研讨会作为用户和开发者之间的讨论要点。在这些会议期间,研究人员收集了用户对系统和使用场景的反馈以及未使用和误用已实现功能的信息。Kautz^[93]也研究了ASD内的参与式设计活动。他的报告表示沟通是通过不断收集用户参与的数据,包括策划游戏、展示工作软件和验收测试来进行的。Lucia等^[101]提供了敏捷需求工程的概述,访谈、头脑风暴和用例分析是他们最重要的启发技巧。Nakki等^[100]将用户看成是想法的来源,并在整个设计过程中将其视为决策者。为此他们引导用户通过与社交机制的在线共创工具参与到整个设计过程中来。考虑到使用环境分析是实现以人为本的过程的一项重要活动,一些研究者建议结构化该过程。他们描述了如何使用一种适当的方式^[102-103]进行使用情景调查^[104]。关于以人为本的设计理念,Maguire^[97]提出了一些遵循[ISO9241-210]标准的不同方法。除了情景调查之外,他还建议完成涉众分析,以探索使用情景。此外,他还提出了调查、访谈、讨论、重点小组、竞争对手分析和用户旅程,以确定用户需求。Maguire建议在创建设计解决方案期间,通过概念设计会议和协同设计研讨会来收集数据。在评估中,他强调用户跟进和可用性测试的使用很有必要。

还有研究致力于扩展敏捷方法,如scrum和XP。Maguire^[97]扩展了用于敏捷开发的HCD框架(ISO9241-210)。在HCD中有4个主要活动,它们是迭代执行的:1)理解和指定使用情景;2)指定用户需求;3)产生设计解决方案;4)根据需求评估设计。Maguire建议使用不同的方法来执行每一项活动。此外,他还建议根据收集到的信息生成具有明确定义的构件。Adikari等^[99]提出了一个基于设计思维、UX设计和ASD这3种方法的框架。真实的世界系统环境可以利用设计思维构建的相关系统探索。他们的报告显示用于重构环境的知识可以进行UX设计和ASD创建产品、系统或服务。情景调查的几项研究(如文献[102]和文献[103])显示情景查询^[88]有助于探索用户观点,并收集有关用户和使用情景的数据。Kautz^[93]研究了客户和用户如何通过参与性设计参与ASD。Kautz专注于研究客户和用户在开发过程中扮演的角色,以及他们如何通过设计和开发中的不同的活动参与进来。Bellucci等^[85]研究了一种基于用户的设计方法,为此他们进行了一项实地研究,以评估用户如何与原型交互。根据他们的发现,原型是通过恒定的反馈促进迭代开发的。此外,Nakki等^[100]在ASD的背景下,致力于研究基于用户开发的应用。所选的潜在用户通过生成创意积极参与创新过程,并通过在线协同开发工具参与开发过程的所有阶段。关于参与性设计,Olsson等^[87]在多案例研究的基础上,建立了QCD

模型。他们认为将定性和定量反馈技术相结合以实现持续的客户验证很重要。

敏捷需求工程环境中的工具被用于需求的沟通、阐述、验证和文档化。用户故事是 ASD 中最常用的工具。Nakki 等^[100]使用了基于需求的用户故事的概念,为此他们收集了用户对特定领域的日常需求和困难。用户在参与需求制定过程中,通过对特性进行讨论和评级,以便对特性进行优先级排

序。Harbers 等^[86]建议使用价值故事研讨会将利益相关者价值嵌入到需求的引出过程中。这种研讨会产生的需求是基于价值的用户故事。除了经典用户故事^[77],Wanderley 等^[94]为用户故事提供一种可视化语言。这种可视化语言支持与用户一起评估需求,并且可以与用户故事可视化编辑器一起使用。

文中还总结了部分文献中为研究敏捷需求工程所采用的工具,如表 3 所列。

表 3 敏捷需求工程研究所采用的工具

Table 3 Agile requirements Engineering research tools statistics

工具	描述	论文	占比/%
用户故事	从用户的角度来描述一个特征,它由书面文本,相关对话和可接受标准组成	[86][87][88][89][91][92][93][95][96][99][100][103][104][105][107]	56
原型	用于评估设计备选方案和交流质量的软件应用程序模型	[83][85][86][92][95][96][97][100][102][101][103][104][105]	41
用例	描述一个动作或事件步骤	[82][86][101][103][105][106][107]	26
场景	问题的文本表示描述了特定上下文中用户和系统之间的交互	[86][96][97][101][102][105]	22
故事卡	用户故事中书面文本和细节的物理表示	[87][92][93][103][107]	22
角色	对代表大部分目标群体的虚构人物的描述。	[89][95][97][102]	15
期望	对指导产品开发并使开发、业务人员和其他利益相关者保持一致的总体目标的抽象描述。	[89][101][103]	11
UML 图	统一建模语言(UML)提供了可视化系统设计的标准	[88][94][99]	11
故事板	以图片形式展示用户工作流程	[96][100][101]	11
任务	一个用户故事被分成更多的任务,这些任务描述了更多的技术要求	[89][96]	7
看板	通过开发团队的工作流程将需求的进度可视化	[89][103]	7
UI 模式	描述了针对重复出现的设计问题的抽象解决方案,并为设计师提供灵感	[99][104]	7
基本用例	描述了用户任务,是用例的简化和通用形式	[99][104]	7
图片	一种视觉表现,如照片和绘画	[85][100]	7
视频	由一系列经过电子处理的图像组成,这些图像在录像中看到并显示在屏幕上	[85][100]	7
思维导图	用来可视化和组织信息的图表	[94][106]	7
UI 规范	描述系统 UI 的书面规范,通过模型、图标等丰富文本的表现形式	[97][102]	

通过对不同类型的准确度(低、中和高)的研究可以为原型分类。Lucia 等^[101]建议使用纸制原型来记录需求,目的是便于在利益相关者和敏捷开发团队之间进行沟通和知识共享。此外,Obendorf 等^[96]说明纸制原型和非正式文档在与用户讨论时非常有用。Blomkvist 等也使用了非正式文档^[102]。Rivero 等^[104]则认为,使用 html 模型在模型驱动的 Web 工程中启动建模过程,一方面是作为指定内容、导航和业务逻辑等特性的基础,另一方面是生成独立于平台的 UI 规范。此外,Nawrocki 等^[105]建议使用原型从用户那里导出测试用例来使测试用例更加可靠。与用户故事相比,用例(UC)通常用于从更有技术性的角度描述系统的行为。Issa 等^[106]建立了一个 UC 模式目录,可以用作功能清单,并设计了一个 UC 元模型。他们的 UC 元模型涉及预期系统的环境、技术、结构、最终和可追溯性等方面。此外,Nawrocki 等^[105]认为 UC 可以重用于用户手动生成和工作估计。Farid^[107]将需求分为功能性和非功能性。功能需求就是敏捷用例,而非功能需求就是松敏捷案例。使用面向特性的“切入点”操作符可以将功能性与非功能性需求连接起来。

场景描述了用户如何在特定情景中与系统交互。为此,往往将它们与角色^[92,99]相结合。Bellucci 等^[85]的研究中,设计者们使用日志来记录基于用户意见的场景,收集用户日记和试用记录。此外,Obendorf 等^[96]在基于场景的可用性工程

方法中,使用场景将设计理念与开发者的技术任务连接起来。

3.3 数据驱动的需求工程

得益于通信技术和存储技术的发展,整个组织和社会的持续数字化,电子商务激增、边缘计算和物联网技术的出现导致了前所未有的高速和异构数据,通常被称为大数据^[108]。在传统需求工程中,需求从利益相关者获得的领域知识中引出,主要依赖于定性数据收集方法(如访谈、研讨会和焦点小组讨论)^[109]。大数据给传统需求工程带来了新的机会,除了领域知识外,还可以将这种来自各种来源的动态数据视为潜在有价值的需求来源。以互补的方式利用传统和新数据源有助于提高软件质量或促进新软件系统的开发。然而,传统的引出技术对于处理这些快速增长的数据而言耗时巨大,也不能与与日俱增的全球利益相关者群体的需求相匹配。

目前已经有许多从静态数据中自动获取需求的方法,它们可以以较低的速度生成很少更新的数据。例如从静态领域知识中提取需求,包括用自然语言编写的文档^[110-111]、本体^[112-113]和各种类型的模型(如业务流程模型^[114]、UML 用例和序列图^[115]);根据已经引出的需求执行特定的需求工程活动,如需求优先级^[116]、自然语言需求的分类^[117]、需求可追溯性的管理^[118]、需求验证^[119]和从自然语言需求生成概念模型^[120];开发工具以提高利益相关者基于静态领域知识或现有需求执行需求工程活动的的能力,如协作需求优先级优化

工具支持^[121]和使用基于规则的推理进行需求协商^[122]。

由于从动态数据中提取需求是较为新颖的研究方向,故相关研究较少,关注这些来自“非刻意”数据源的动态数据有4个主要优势(“非刻意”的数据源指那些专为需求工程收集数据的渠道)。第一,动态数据驱动的需求工程促进了数据的二次使用,从而消除了专门为需求工程收集数据的需要,增强了可伸缩性。Groen等^[123]证明了自动用户审查分析的有效性和可伸缩性,并强调了自动化分析大量动态数据的必要性,以支持连续的需求工程。第二,“非刻意”的数据源包括与新系统需求相关的数据,这些数据是很难被发掘的,因为它们是利用“非刻意”数据源从全球的利益相关者那里收集得来,而依靠传统方法提取需求的组织无法获取这类利益相关者的数据^[124]。对于当前软件系统不支持的一些需求,也可以通过提高客户满意度、降低成本以及优化操作的形式带来商业价值^[125]。第三,关注动态数据可以捕获最新的用户需求,从而实现及时和有效的运营决策。最后,来自“非刻意”数据源的动态

数据是机器可读的,这有助于自动化和连续的需求工程。这种恰当的需求提取方法通过从各种数据源中提取实时商业热点和知识,在快速增长的市场中获得了新的机会和竞争优势。

基于人群的需求工程(CrowdRE)是一个很好的例子,它利用了来自非刻意数据源的动态数据。CrowdRE的重点是利用基于机器学习和自然语言处理的各种技术,从人群用户的明确反馈(如应用评论和社交媒体数据)中得出需求^[126]。Genc-Nayebi等^[127]挖掘了来自移动应用商店的用户评论并进行系统审查,以确定挖掘应用用户评论的现有解决方案和挑战,并提出未来的研究方向。他们专注于分析用于用户评论的特定数据挖掘技术,领域适应方法,制定用户评论有效性的评估标准,筛出无用评论的技术以及应用程序特性。Martin等^[128]分析应用程序评论以提取软件工程的技术性和非技术性特性。Tavakoli等^[129]对通过移动应用程序提取有用的软件开发信息的技术和工具进行了系统的综述。表4列出了部分论文中自动化需求提取的数据来源。

表4 自动化需求提取的动态数据来源

Table 4 Dynamic data source for automated requirements extraction

数据来源	描述	文献	占比/%
在线评论	包括 app 评论、专家点评和用户在线评论,大多数研究数据来源于 app 评论(占 62 左右)	[109][110][111][112][113][114][115][116][117][118][119][120][121][122][133][132][123][126][124][125][126][127][128][129][130][131][132][133][134][135][136][137][139][140][141]	62
博客	包括 twitter、Facebook 和微博,部分研究仅提取文本数据,余下研究还提取了额外元数据	[138][140][141][142][143][144][145]	14
在线论坛	包括论坛 SourceForge 上的密码安全工程项目,Stack-Overflow 的问答,开源论坛 SugarCRM, SecondLife, OpenOffice, Reddit	[146][147][148][149]	7
软件库	包括 Apache OpenOffice, Android OS, GitHub 和 JIRA	[150][151][152]	5
软件/app 产品描述	包括 Softpedia 上的产品描述、更新日志	[153][154]	3
传感器读数	包括手持研磨机的传感器和手机传感器	[156][157]	3
用途数据	一个基于名为 MyReview 的开源 Web 应用程序修改的在线图书馆系统	[158][159]	3
邮件列表	包括 Apache Common User List 和开源软件邮件列表	[115][116]	3

数据驱动的需求工程的另一种方法是在需求工程中使用过程挖掘。过程挖掘是一种基于证据的方法,它主要从事件日志、已发现的模型和预定义的过程模型中推断出有价值的与过程相关的意见。过程挖掘可以分为3种类型:过程发现、一致性检查和过程增强^[130]。Ghasemi等对面向目标的过程建模进行了系统综述,相关的研究分为3个领域:1)目标建模和需求引出;2)意图挖掘(即发掘过程模型的发展趋势,而不仅是正在活动的过程模型);3)关键性能指标(即监控目标的手段)^[130]。Zowghi等^[131]对用于引出需求的技术、方法和工具进行了全面的调查。但他们的研究集中在传统的、利益相关者驱动的需求引出方法上。Arruda等^[132]确定了开发大数据应用程序所需的过程和类型以及大数据应用程序中与需求工程相关的困难和挑战,讨论了可用的大数据应用程序开发需求工程解决方案,并提出了未来的研究方向。

分类自动需求提取的背景涉及以下3个任务:1)过滤与需求无关的数据;2)根据与不同利益相关者群体的特性对文本进行分类;3)将文本分类为不同类别的技术问题,如错误报告和功能需求。分类任务一般使用基于规则的方法或机器学习来解决,主要在有监督的机器学习完成。尽管有监督的机器学习可以在定义明确的分类任务中实现较高的预测

性能,但它需要分析足够数量的人工标注数据。因此许多研究都将人为标注的数据设置为预定义类。不过由于大量的噪声和自然语言特有的模棱两可的性质,标注数据费时费力并且容易出错^[133]。

主动学习和半监督机器学习可以降低大量数据标注成本。Dhinakaran等^[133]使用主动学习策略训练的分类器在将应用程序评论分类为特征请求、错误报告、用户评级或用户体验方面,比在随机选择的数据集上被动训练的基线分类器表现更好。Deocadez等^[134]证明了3种半监督算法(自我训练、RASCO和Rel-RASCO)和4个基础分类器在将应用程序审查分类为功能或非功能需求时,取得了与经典有监督机器学习相当的预测性能。

在使用监督学习时需要解决的另一个问题是,来自用户评论的数据源很大比例是无意义或无关的数据。从这种数据源中提取需求无异于“大海捞针”^[135]。这就导致了无意义和无关数据相对于有意义和相关数据的类分布高度不均匀。底层的类分布很大程度上影响了基于机器学习的分类器的性能^[136-137]。文献[138]中发现分布不均匀的类的精确度、召回率和F1指标比分布较好的类差。鉴于在现实中大多数与需求相关的类没有相似的表现,这个问题需要在未来的研究中得到解决。一个可能的解决方案是在训练集中使用不同的

采样技术,如合成少数过采样技术(SMOTE),以使用较少的观测增加类中的实例数量^[139-140]。

情景化(Contextualization)是另一种可能的解决方案,它通过筛选无意义的和无关的数据来实现。在进行角粒度分类或聚类之前,一些研究^[141-144]使用了监督分类,通过过滤掉有噪声的数据,提高分类或聚类的精度。它还自动为需求工程师丢弃无意义数据,帮助确定与需求相关的数据,并支持向组织中的特定利益相关者分发数据^[145]。由于语境化可以减少要进一步处理的数据量,因此它缓解了大数据的规模问题。

表 5 分类和聚类方法汇总

Table 5 Summary by classification and clustering methods

分类方法	描述
机器学习	1. 监督学习 ^[109,111,113,115-118,120,124-127,129-131,133,135,139-145] 2. 朴素贝叶斯、多项式贝叶斯、支持向量机和逻辑回归 ^[111,143,156] 3. 不确定度采样 ^[133] 4. 半监督学习 ^[134]
基于规则分类	1. 基于规则分类 ^[125,126,129,135,141-143,148-149,155] 2. 语言模式、标签、上下文以及句式分析 ^[123,141,148-149,155]
面向模型	1. 面向用户的概念建模对用户偏好进行分类排序 ^[150] ,转化概念模型中排名靠前的用户偏好为 <i>i*</i> 目标模型,以解释早期系统需求 ^[149] 2. Q-rapids 是一个数据驱动的敏捷软件开发框架,用于快速提取质量需求 ^[147-148]
聚类方法	描述
主题模型	1. LDA(隐迪利克雷分配模型) ^[135,138,140,154] ,关键字扩展改进的 LDA ^[144] ,分层 LDA ^[139] 2. BTM(Biterm Topic Model) ^[142,153]
传统聚类	1. K-means 聚类算法 ^[121,132,140-141,144,149] 2. Bagging 聚类算法(结合 EM、K-means 和 Mtree 聚类算法) ^[160]
基于规则聚类	1. 文献 ^[161] 提出了一种无监督聚类算法,该算法使用预定义的关键字列表、词向量表示和文本模式从用户反馈中提取上下文

此外,机器生成的数据在一些研究中被用作需求来源。Liang 等^[145]建议使用先验 *m* 算法,从移动用户的实时行为挖掘出来的行为模式中推断出具有情景感知的需求,其分析结果可以提供满足相关需求的用户解决方案。Voet 等^[146]提出了一种利用有监督学习对目标相关的使用元素状态进行分类的方法,并基于与最佳使用模式的偏差来推断需求,这种偏差可以通过手动分析或无监督聚类来检测。IoT 数据是大数据生成的主要源头之一,因此迫切需要开发一个框架来从 IoT 数据中引出需求。语义技术可能是一种很有前途的解决方案,语义表示形式^[147]的原始数据可以帮助机器理解数据的意义,还有助于从大量异构 IoT 数据中自动引出需求。

目前与数据驱动的需求工程相关的研究中,与过程生成或机器生成的数据源相比,来自人类活动产生的数据目前明显占主导地位。因此,用于数据处理的技术是基于自然语言处理的,使用机器学习进行分类和聚类则普遍用于完全自动化引出需求,而不是将其与传统的涉及利益相关者的方法相结合。未来一个可能的研究方向是使用和分析更广泛的非人类活动产生的数据类型,以及要研究自动数据融合和情景方法来集成、处理和分析大量的异构数据源,以引出需求。语义技术可以成为解决大数据的多样性和规模问题的一个很有前途的解决方案。实现实时数据处理和分析,高速发掘大数据的连续需求也是一个研究方向。此外,每个解决方案都需要对照传统需求进行评估,以说服从业者现实中采纳它们。还需要对原始数据的内容和质量进行进一步的改进以引出更详细的需求。最后,使情景感知能够捕获随时间动态变化的需求也非常重要。

Cleland-Huang 等^[144]提出了自动论坛管理(AFM)系统,该系统使用稳定球形 *k* 均值算法(SPK)挖掘来自开源论坛中讨论线程的特性需求。Sun 等^[140]使用推荐的聚类数量(RCN)来确定聚类的最佳数量。

表 5 描述了所选研究中使用的自动化需求获取方法的描述性统计数据。对于分类,最常用的方法是机器学习(60%),其次是基于规则的分类(17%)和面向模型的方法(6%)。对于聚类,主题模型(16%)最为常用,其次是更传统的聚类技术(13%)和基于规则的无监督聚类(2%)。

4 需求工程的困难与挑战

需求工程(RE)的目的是引出、分析和规范需求,它能明确地反映软件系统的预期目的,考虑并协调所有利益相关者的观点。精确且一致的需求有助于优化系统开发的成本效益^[148],RE 是生产力和产品质量的决定因素^[149]。然而,由于工业界的各种影响,RE 仍然是一个非常复杂的学科。由于在项目立项时,许多方面还不清楚,故这一过程本身是由不确定性驱动的^[150]。项目配置会影响 RE 方法和工具的选择,这在其他软件工程学科中并不存在。这种情况使得这一学科不可能标准化,也不可能全面的解决方案。

为查明工业界中软件在需求工程方面所面临的困难,很多研究人员进行了大量的调查。其中最著名的是 Standish 集团的 Chaos 报告,它特别关注项目失败的根本原因,其中大多数与 RE 直接相关,比如缺少用户的参与等。不过 Chaos 报告的设计存在严重缺陷,对结果的有效性产生了负面影响^[151]。Buschermöhle 等^[152]在德国开展的 Success 研究对德国公司进行了类似的调查,具备详细且可重复的研究设计。尽管如此,这两项调查都只查明了失败的项目和总体软件过程级别的一般原因。Kamata 等^[153]的研究也聚焦于类似的重点,但仅限于 RE 领域。他们分析了 IEEE 软件需求规范 Std. 830-1998(IEEE 1998)中单个部分对项目成功的关键性。

上述研究的重点并不支持对工业界中近况和 RE 问题的调查。Nikula 等^[154]开展了对芬兰中小企业组织层面的调查。他们根据调查发现,推断出了改进的目标,例如优化知识转移。Staples 等^[155]调查了工业界不愿进行软件开发过程

改进的原因。他们发现了公司不采用规范的改进解决方案的不同原因,如 CMMI 和相关框架(专注于公司的评估和基准化分析,而不是问题驱动的改进,见 Napier 等^[156]和 Pettersson 等^[157]的研究)。其不愿建立规范的改进框架的原因是:改进开发流程并不能给这些规模较小的公司带来明显的好处。

Beecham 等^[158]开展了一项在实际环境中发现相关问题的调查。他们的调查报告印证了 Hsia 等^[159]所讨论的问题,还调查了一组关键的组织性和项目特定的问题,如沟通问题、不适用的技术或模糊的需求。Solemon 等^[162]的调查报告显示了马来西亚软件公司在 RE 实践方面的一些问题。Liu 等^[163]在中国进行了关于 RE 实践方面问题的调查。Verner 等^[164]在澳大利亚和美国进行了一项调查。他们专注于发掘 RE 中的成功因素,并发现良好的需求、客户/用户的积极参与和有效的需求管理是项目成功的最佳因素。Asaad^[165]发起了一项针对 GCC(海湾阿拉伯国家合作委员会)RE 应用情况的详细调查,该调查涉及沙特、阿联酋、卡塔尔、巴林和也门超过 163 家软件公司,详细统计了这些公司在需求获取方式、需求获取耗时、项目立项、项目完成情况、项目规模和项目范围界定方面的表现情况。

两年一度的 NaPiRE(Naming the Pain in Requirements Engineering)报告是了解软件从业者在 RE 方面遇到的各种困难的理想参考,该报告通常涉及 10 余个国家的数百家企业,由 20 余名来自不同国家的研究人员负责运营,旨在建立一个关于 RE 实际问题和需求现状的共享知识库,从长远看,它甚至能帮助建立一套涵盖特定现象中广泛背景因素的 RE 理论。根据近年的 NaPiRE 报告^[166]来看,目前出现频次最高的 5 个 RE 问题分别是需求实现不完全、项目团队与客户缺乏交流、开发中途改变项目目标、需求不足且过于抽象和时间不足,而导致项目失败概率最高的 5 个 RE 问题是客户未提供有效支持、改变项目目标、项目团队与客户缺乏交流、项目团队内部缺乏交流和需求实现不完全。诱发 RE 问题的五大主要因素是缺乏时间、缺乏有 RE 管理经验的团队成员、项目组和客户缺乏交流、需求过于抽象和商业需要变化。此外,报告还对每个导致 RE 问题的因素进行细致分析,例如项目若出现需求实现不完全的情况,则来自项目本身或项目组的影响最大(43%),其中超时、超支和低效率的影响分别达到 10.34%,8.05%和 6.9%。

结束语 本文综述了需求工程的发展历程,包括从需求工程方法学开始,到面向对象建模,再到基于本体和面相特征的需求工程,以及与机器学习相关的自动化需求提取技术。随后重点介绍了 3 种需求工程相关的研究方向,包括需求工程中采用的自然语言处理方式和部分语言类型,敏捷需求工程的发展历程以及近年提出的工具和方法,数据驱动的需求工程的概念、必要性、过程和近年主要采取的方法和实践。最后基于各国关于需求工程开展情况的报告,分析了需求工程近年遭遇的困难和挑战。

就敏捷需求工程而言,本文共确定了 27 项相关研究并进行了分析,根据发表渠道和研究方法对研究结果进行定量分类,纳入的研究发表于 2007 年至 2018 年。基于对纳入研究的定性分析得出,相关研究并没有很好地提出在 ASD 中建立对用户观点的通用解释方法。并且只有少数论文体现了

ASD 中用户观点的存在。这些研究表明,敏捷需求工程在用户和利益相关者的直接参与方面存在许多问题。一种思路是将 4 种方法(以人为本的设计、设计思维、语境探究和参与式设计)整合到 ASD 中,旨在增加对用户需求的理解。此外,本文确定了可收集 RE 数据以用于 ASD 的多种不同方法。还为敏捷 RE 中使用的需求文档确定了以下关键工具:用户故事、原型、用例、场景和故事卡。工业界可以利用这些发现为他们的开发过程使用正确的工具组合。关于 NFR(非功能性需求),一方面有多种不同的方法来处理 NFR,另一方面,评估这些需求是个很大的难题(如 UX 指标、安全策略)。总而言之,还需要更多的实际研究,使用各种项目设置(如不同的敏捷方法、项目成员的规模或差异程度)对敏捷 RE 进行评估。此外还可以得出,敏捷 RE 方法之间存在异质性,侧重于用户和利益相关者的参与。未来的研究可能会尝试建立专门处理利益相关者和用户参与 ASD 的常用过程模型。为此,可以创建一个元模型,在更高层次上描述敏捷 RE 的影响参数。此外,必须找到适当的方法在项目成员和利益相关者之间建立关于用户观点的通用解释方法。本文对敏捷 RE 研究的回顾还表明,由于在实践中缺乏适当的指导方针,需求管理主题需要进一步的实验评估。

就数据驱动的需求工程而言,与过程导致的和机器生成的数据源相比,网络用户数据源明显占主导地位。因此,用于数据处理的技术通常基于自然语言处理,而使用机器学习进行分类和聚类是很普遍的。

相关研究提出的方法的主要目的是使需求提取过程完全自动化,而不是将其与传统利益相关者参与相结合。此外大多数研究都考虑了功能和非功能(即质量)要求。目前关于获取的数据在系统开发或演进中的完整性和准备情况的最终结果比较有限,即大多数研究都获得了与需求内容相关的一些信息,一些研究针对的是核心功能或质量识别。就特征而言,只有少数研究达到了高水平的内容要求。最后,大多数研究评估了实验环境中的结果,这表明该方法在实际需求工程环境中的实施程度相当低。

所得结果为今后的工作提供了若干方向。一个可能的方向涉及对机器生成的数据类型采取更广泛的使用和分析的调查。此外,需要研究自动数据融合和上下文方法,以集成、处理和分析大量异构数据源来提取需求。语义分析技术可以成为解决大数据种类和数量问题的有效解决方案。另外还可以采取更好的实时数据处理和分析方法,以便快速地从大数据中持续获取需求。此外,每个提议的解决方案都需要根据传统要求进行评估,以说服从业者在工程项目中实施。为满足这类全面且详细传统要求而引出的数据的内容和质量也需要进一步改进。最后,一个非常有前景的研究方向是运用上下文感知以捕获随时间动态变化的需求。

就需求工程的自然语言处理而言,将需求转换为概念模型是 SDLC(软件开发生命周期)中一项重要且具有挑战性的任务。尽管相关文献中存在许多自动化该过程的方法,但似乎还没有实用可行的自动化解决方案。本文对共计 44 项相关研究的方法、功能、使用的数据集、评估方法、生成的模型类型和支持的语言进行了评估。

评估结果显示重点问题在于如何提出一种概念识别方法将软件需求和概念模型联系起来。而且,绝大多数研究涉及

生成英语需求的概念模型。它们为分析英语文档的软件工程领域做出了重要贡献。但由于涉及更多非英语文档能接触到更多的用户,因此,在需求分析中为多语言提供通用方法可确保概念识别研究的广泛性和可访问性。

另一个相当大的限制是当前的研究通常只关注生成UML类图。但是,有14种类型的UML类图可用于表示软件的不同方面的特征。因此,使用单独的图表描述完整的设计可以使整体设计更易于理解并保持模块化组件的完整性。此外,尽管关系识别是构建完整和一致的概念模型的最关键任务,但生成模型的组件之间的关系依然没有被正确识别。无法确定组件关系可能会限制设计和实施阶段之间的可追溯性。为了克服这些缺陷,建议设计一个具有更具体的转换规则的算法。

在项目管理方面,一个未来的可能研究方向是根据项目和项目组情况,帮助项目组制定最容易出现的RE问题的相应的预防计划。例如采用敏捷开发的小型团队中,最可能出现的RE问题是需求实现不完全,为减少此类问题出现的频率,可以采取以下方法:1)检查团队成员的资格和经验,根据需要提供培训,特别是避免过于抽象的文档;2)调整投资管理组合,以避免在极端的时间压力下承担项目,尽量避免与缺乏商业嗅觉的利益相关者合作;3)评估和提高其需求引出技术的效率;4)改进项目完整性检查,例如,计划驱动的团队可以根据RE标准将需求检查制度化,而敏捷组织可以考虑引入“准备就绪定义”(DoR),这在敏捷项目中很常用,它可以避免开发团队在没有明确定义完成标准的功能点上开始工作。

参 考 文 献

- [1] ELIZABETH H, KENNETH J, JEREMY D. Requirements Engineering[M]. Berlin: Springer, 2004: 1-3.
- [2] LI M S, WANG Q. The research status of requirements engineering[J]. Chinese users of computer, 1999(47): 1-5.
- [3] LANO R J. A structured approach for operational concept formulation[C]// Tutorial: System and Software Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1990: 48-57.
- [4] SUTCLIFFE A. The domain theory for requirement engineering[J]. IEEE Transactions on Software Engineering, 1998, 24(3): 174-195.
- [5] MAIDEN N A M. Requirements critiquing using domain abstractions[C]// Proceedings of 1st International Conference on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994: 184-193.
- [6] ALFORD M. Attacking requirements complexity using a separation of concerns[C]// Proceedings of 1st International Conference on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994: 2-5.
- [7] CHOU S, CHUNG C. An OOA model with system function specifications[C]// Proceedings of 1st International Conference on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994: 16-23.
- [8] ECKERT G. Types, classes and collection in object-oriented analysis[C]// Proceedings of 1st International Conference on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994: 32-39.
- [9] DRAKE J M. System bounding issues for analysis[C]// Proceedings of 1st International Conference on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994: 24-31.
- [10] GOTEL O, FINKELSTEIN A. An analysis of the requirements traceability problem[C]// Proceedings of 1st International Conference on Requirements Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1994: 94-101.
- [11] LU R L, JIN Z, CHEN G. The analysis of ontology-oriented requirements[J]. Chinese Journal of Software, 2000, 23(5): 486-492.
- [12] JIN Z. Ontology based automatic requirements retrieving[J]. Chinese Journal of Computer Science, 2000, 23(5): 486-492.
- [13] KANG K C, COHEN S G, HESS J A, et al. Feature-Oriented Domain Analysis (FODA) Feasibility Study: Technical Report CMUSEI-90-TR-21[R]. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1990.
- [14] GOMAA H V, SUGUMARAN H V, BOSCH C, et al. A Prototype Domain Modeling Environment for Reusable Software Architectures[C]// Proceedings of the Third International Conference on the Software Reuse. IEEE Computer Society, 1994: 74-83.
- [15] ROTAK S, PIECZYNSKI A. Employing fuzzy logic in feature diagrams to model variability in software productlines[C]// Proceedings of the 10th International Conference and Workshop on the Engineering of Computer-Based Systems, 2003: 305-311.
- [16] PAPAIEWSKI B D, SCHRODER-PREIKSCHAT W. Variability management with feature models[J]. Science of Computer Programming, 2004, 53(3): 333-352.
- [17] CZAMECKI K, HELSEN S, EISENECKER U. Formalizing cardinality-based feature models and their specialization[J]. Software Process: Improvement and Practice, 2005, 10(1): 7-29.
- [18] VON DER MALEN T, LICHTER H. RequiLine: A requirements engineering tool for software productlines[J]. Software Product-Family Engineering, 2004, 3(14): 168-180.
- [19] MEZINI M, OSTERMANN K. Variability management with feature-oriented programming and aspects[C]// Proceedings of the 12th Int'l Symp. on Foundations of Software Engineering. New York: ACM Press, 2004: 127-136.
- [20] HARTMANN H, TREW T. Using feature diagrams with context variability to model multiple product lines for software supply chains[C]// Proceedings of the 12th International Conference on Software Product Line Conference. 2008: 12-21.
- [21] CZAMECKI K, ANTKIEWICZ M, KIM C H P, et al. fimp and fmp2rsm Eclipse plugins for modeling features using model templates[C]// Proceedings of the International Conference on Object-Oriented Programming, Systems, Languages, and Applications, 2005.
- [22] HAMMOUDA I, HAUTAMAKI J, PUSSINEN M, et al. Managing variability using heterogeneous feature variation patterns[C]// Proceedings of the 12th International Conference on Fundamental Approaches to Software Engineering, 2005: 145-159.
- [23] LEE J, MUTHIG D. Feature-Oriented variability management in product line engineering[R]. Communications of the ACM-Software Product Line, 2006, 49(12): 55-59.
- [24] DHUNGANA D. Integrated variability modeling of features and

- architecture in software product line engineering[C]// Proceedings of the 21st International Conference on Automated Software Engineering, 2006:327-330.
- [25] DAG J N O, REGNELL B, CARLSHAMRE P, et al. A Feasibility Study of Automated Natural Language Requirements Analysis in Market-Driven Development[M]. Requirements Engineering, 2002:20-33.
- [26] EDUARD C G. The Crowd in Requirements Engineering[J]. IEEE Software, 2017, 34(2):44-52.
- [27] ANDREAS B. Requirements Engineering Method for Infrastructure Automation and Cloud Projects[C]// IEEE 27th International Requirements Engineering Conference (RE). IEEE Computer Society, 2019.
- [28] FERRARI A, DELL'ORLETTA F, ESULI A, et al. Natural language requirements processing: A 4D vision[J]. IEEE Software, 2017, 34(6):28-35.
- [29] JONAS P W, JAM G, ANDREAS A, et al. Optimizing for Recall in Automatic Requirements Classification: An Empirical Study[C]// IEEE 27th International Requirements Engineering Conference (RE). IEEE Computer Society, 2019.
- [30] CLELAND-HUANG J, SETTIMI R, ZOU X, et al. Automated classification of non-functional requirements[M]. Requirements Engineering, 2007, 12(2):103-120.
- [31] KURTANOVIC Z, MAALEJ W. Automatically classifying functional and non-functional requirements using supervised machine learning[C]// IEEE International Requirements Engineering Conference (RE). 2017:490-495.
- [32] BINKHONAIN M, ZHAO L. A review of machine learning algorithms for identification and classification of non-functional requirements[J]. Expert Systems with Applications, 2019, 6(2):88-89.
- [33] ABAD Z S H, KARRAS O, GHAZI P, et al. What works better? A study of classifying requirements[C]// IEEE International Requirements Engineering Conference. 2017:496-501.
- [34] MAALEJ W, KURTANOVIC Z, NABIL H, et al. On the automatic classification of app reviews[J]. Requirements Engineering, 2016, 21(3):311-331.
- [35] CHEN N, LIN J, HOI S C, et al. AR-miner: Mining informative reviews for developers from mobile app marketplace[C]// International Conference on Software Engineering. 2014:767-778.
- [36] PRESSMAN R S, MAXIM B. Software Engineering: A Practitioner's Approach[M]. McGraw-Hill Education, 2014.
- [37] SAGAR V B, ABIRAMI S. Conceptual modeling of natural language functional requirements[J]. Journal of System & Software, 2014(88):25-41.
- [38] JOHNSON J, HENDERSON A. Conceptual models; begin by designing what to design[J]. Interactions, 2002, 9(1):25-32.
- [39] DAWOOD O, SAHRAOUI A. Toward requirements and design traceability using Natural Language Processing[J]. European Journal of Engineering Research and Science, 2018, 3(7):42-49.
- [40] BOZYIĞIT F, AKTAS Ö, KILINÇ D. Automatic concept identification of software requirements in Turkish[J]. Turkish Journal of Electronics and Engineering Research, 2019, 27(1):453-470.
- [41] OMAR M, BARYANNIS G. Semi-automated development of conceptual models from natural language text[J]. Data Knowledge Engineering, 2020, 18(1):127.
- [42] ELALLAOUI M, NAFIL K, TOUAHNI R. Automatic transformation of user stories into UML use case diagrams using NLP techniques[J]. Procedia Computer Science, 2018, 130:42-49.
- [43] QUIRCHMAYR T, PAECH B, KOHL R, et al. Semi-automatic rulebased domain terminology and software feature-relevant information extraction from natural language user manuals[J]. Empirical Software Engineering, 2018, 23(6):3630-3683.
- [44] REDDY C R M, GEETHA D E, RAO R R, et al. Transformation of user interface to activity models and assessing performance of WA/WS[J]. Journal of Software Engineering Applications, 2019, 12(5):101-127.
- [45] SADAT F, YOSHIKAWA M, UEMURA S. Cross-language information retrieval using multiple resources and combinations for query expansion[J]. Advances in Information Systems, 2002, 17(5):114-122.
- [46] MONTES A, PACHECO H, ESTRADA H, et al. Conceptual model generation from requirements model: A Natural language processing approach[C]// Natural Language and Information Systems. 2008:325-326.
- [47] LIU D, SUBRAMANIAM K, EBERLEIN A, et al. Natural language requirements analysis and class model generation using UCDA[C]// Innovations in Applied Artificial Intelligence. 2004:295-304.
- [48] JAIWAI M, SAMMAPUN U. Extracting UML class diagrams from software requirements in Thai using NLP[C]// 14th International Joint Conference on Computer Science and Software Engineering (JCSSE). 2017:1-5.
- [49] MICH L. NL-OOPS: from natural language to object oriented requirements using the natural language processing system LOLITA[J]. Natural Language Engineering, 1996, 2(2):161-187.
- [50] OVERMYER S P, BENOIT L, OWEN R. Conceptual modeling through linguistic analysis using LIDA[C]// Proceedings of the 23rd International Conference on Software Engineering. 2001:401-410.
- [51] CAPUCHINO A M, JURISTO N, VAN DE RIET R P. Formal justification in object oriented modelling: a linguistic approach[J] Data Knowledge Engineering, 2000, 33(1):25-47.
- [52] WAHONO R S, FAR B H. A framework for object identification and refinement process in object-oriented analysis and design[C]// IEEE International Conference on Cognitive Informatics. IEEE, 2002:351-360.
- [53] BOZYIĞIT F, AKTAS Ö, KILINÇ D. Linking software requirements and conceptual models: A systematic literature review[J]. Engineering Science and Technology, 2021, 21(2):78-146.
- [54] HEINE B, NARROG H. The Oxford handbook of linguistic analysis[M]. Oxford Handbooks in Linguistic, 2015.
- [55] INSFRÁN E, PASTOR O, WIERINGA R. Requirements engineering-based conceptual modelling[J]. Requirements Engineering, 2002, 7(2):61-72.
- [56] PEREZ-GONZALEZ H G, KALITA J K. GOOAL: A graphic object-oriented analysis laboratory[C]// Companion of the 17th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications. 2002:38-39.
- [57] HARMAIN H M, GAIZAUSKAS R. CM-Builder: a natural language-based case tool for object-oriented analysis[J] Automated

- Software Engineering, 2003, 10(2):157-181.
- [58] YUE T, BRIAND L C, LABICHE Y. An automated approach to transform use cases into activity diagrams[J]. Modelling Foundations Applications, 2010; 337-353.
- [59] SALBRECHTER A, MAYR H, KOP C. Mapping pre-designed business process models to UML[C]// Proceedings of the Eight IASTED International Conference on Software Engineering and Applications, 2004; 400-405.
- [60] CYSNEIROS L M, LEITE J C S. Nonfunctional requirements: from elicitation to conceptual models[J]. IEEE Trans. Software Engineering, 2004, 30(5):328-350.
- [61] AMBRIOLA V, GERVASI V. On the systematic analysis of natural language requirements with CIRCE[J]. Journal of Automated Software Engineering, 2006(3):107-167.
- [62] BAJWA I S, SAMAD A, MUMTAZ S. Object oriented software modelling using NLP based knowledge extraction[J]. European Journal of Engineering Research and Science, 2009(35):22-33.
- [63] BRAMBILLA M. From requirements to implementation of ad-hoc social web applications: an empirical pattern-based approach[J]. IET Software, 2012, 6(2):114-126.
- [64] SHINDE S K, BHOJANE V, MAHAJAN P. NLP based object oriented analysis and design from requirement specification[J]. Journal of Computer, 2012(47):30-34.
- [65] TAKEUCHI H, NONAKA L, TAKEUCHI H. The new product development game[J]. Harvard Business Review, 1986(64):137-146.
- [66] KRUCHTEN P. The Rational Unified Process[R]. 1st Edition, Addison-Wesley, 1998.
- [67] KRUCHTEN P. The Rational Unified Process: An Introduction [R]. 3rd ed, Addison-Wesley, 2004.
- [68] SCHWABER K. SCRUM Development Process, J. Sutherland, C. Casanave[M]. London: Business Object Design and Implementation, 1997.
- [69] SCHWABER K. Agile Project Management with Scrum[M]. Microsoft, 2004.
- [70] BECK K. Extreme Programming Explained: Embrace Change [M]. Addison-Wesley, 2000.
- [71] PALMER S R, FELSING M. A Practical Guide to Feature-Driven Development[M]. Pearson Education, 2001.
- [72] ANDERSON D J. Making the Business Case for Agile Management - Simplifying the Complex System of Software Engineering [C]// Motorola S3 Symposium, 2004; 1-13.
- [73] ANDERSON D J. Kanban - Successful Evolutionary Change for your Technology Business[M]. Blue Hole Press, 2010.
- [74] BECK K, BEEDLE M, VAN BENNEKUM A, et al. Manifesto for Agile Software Development[EB/OL]. <http://www.agilemanifesto.org/2016>, 2001.
- [75] SCHON E M, ESCALONA M, THOMASCHESKI J. Agile Values and Their Implementation in Practice[J]. International Journal of Interactive Multimedia and Artificial Intelligence, 2015(3):61-65.
- [76] COHN M. User Stories Applied: For Agile Software Development[M]. 2004.
- [77] PATTON J. User Story Mapping: Discover the Whole Story, Build the Right Product[J]. First edit, O'Reilly, 2014, 24(3):214-261.
- [78] HUDSON W. User stories don't help users[M]. Interactions, 2013(20):50-53.
- [79] SCHON E, WINTER D, UHLENBROK J, et al. Enterprise Experience into the Integration of Human-Centered Design and Kanban[C]// Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT 2016). Lisbon, Portugal, 2016; 133-140.
- [80] HUMBLE J, FARLEY D. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation [M]. Pearson Education, Inc, Boston, MA, 2010.
- [81] LARUSDOTTIR M K, BJARNADOTTIR E R, GULLIKSEN J. The Focus on Usability in Testing Practices in Industry[C]// Proceedings of Second IFIP TC 13 Symposium, HCIS 2010. Held as Part of WCC 2010, 2010; 98-109.
- [82] BJARNASON E, UNTERKALMSTEINER M, BORG M, et al. Multi-case study of agile requirements engineering and the use of test cases as requirements[J]. Information and Software Technology, 2015(77):61-79.
- [83] PAETSCH F, EBERLEIN A, MAURER F. Requirements engineering and agile software development, WET ICE 2003[C]// Proceedings Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003; 1-6.
- [84] SILLITTI A, SUCCI G, ALBERT O. Requirements engineering for agile methods[J]. Engineering and Managing Software Requirements, 2005; 309-326.
- [85] BELLUCCI A, JACUCCI G, KOTKAVUORI V, et al. Extreme-Co-design: Prototyping with and by the User for Appropriation of Web-connected Tags[J]. International Symposium on End User Development, 2015, 8(3):109-124.
- [86] HARBERS M, DETWEILER C, NEERINCX M A. Requirements Engineering; Foundation for Software Quality [M]. Cham; Springer International Publishing, 2015.
- [87] OLSSON H H, BOSCH J. Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation[J]. LNBI, 2015(210):154-166.
- [88] DRAGICEVIC S, CELAR S, NOVAK L. Use of Method for Elicitation, Documentation, and validation of Software User Requirements (MEDoV) in Agile Software Development Projects [C]// 2014 Sixth International Conference on Computational Intelligence, Communication Systems and Networks, IEEE, 2014; 65-70.
- [89] LISKIN O, SCHNEIDER K, FAGERHOLM F, et al. Understanding the role of requirements artifacts in kanban[C]// Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2014). ACM Press, New York, New York, USA, 2014; 56-63.
- [90] RAMESH B, CAO L, BASKERVILLE R. Agile requirements engineering practices and challenges: an empirical study[J]. Information Systems Journal, 2010(20):449-480.
- [91] LEE S H, KO I H, KANG S, et al. A Usability-Pattern-Based Requirements Analysis Method to Bridge the Gap between User Tasks and Application Features[C]// 2010 IEEE 34th Annual Computer Software and Applications Conference. IEEE, 2010; 317-326.

- [92] BOURIMI M, BARTH T, HAAKE J M, et al. AFFINE for enforcing earlier consideration of NFRs and human factors when building sociotechnical systems following agile methodologies [C] // International Conference on Human-Centred Software Engineering. Berlin: Springer, 2010; 182-189.
- [93] KAUTZ K. Participatory Design Activities and Agile Software Development [C] // IFIP WG 8. 2/8. 6 International Working Conference, 2010; 303-316.
- [94] WANDERLEY F, SILVA A, ARAUJO J, et al. SnapMind: A framework to support consistency and validation of model-based requirements in agile development [C] // 2014 IEEE 4th International Model-Driven Requirements Engineering Workshop (MDRE). IEEE, 2014; 47-56.
- [95] KAMTHAN P. On the interaction between socio-technical dimensions in the next generation of requirements engineering [J]. International Journal of Software Engineering and Its Applications, 2013(7); 183-196.
- [96] OBENDORF H, FINCK M. Scenario-based usability engineering techniques in agile development processes [C] // Proceeding of the Twenty-Sixth Annual CHI Conference Extended Abstracts on Human Factors in Computing Systems-CHIO8. ACM Press, New York, USA, 2008; 21-29.
- [97] MAGUIRE M. Using human factors standards to support user experience and agile design [C] // Proceedings - International Conference (UAHCI 2013). Held as Part of HCI International 2013, Las Vegas, NV, USA, 2013.
- [98] BEYER H, HOLTZBLATT K. Contextual design defining customer-centered systems [M]. Morgan Kaufmann, San Francisco, Calif, 1998.
- [99] ADIKARI S, MCDONALD C, CAMPBELL J. Reframed contexts: Design thinking for agile user experience design [J]. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2013, 6(1); 3-12.
- [100] NAKKI P, KOSKELA K, PIKKARAINEN M. Practical model for user-driven innovation in agile software development [C] // Proceedings of the 2011 17th International Conference on Concurrent Enterprising (ICE 2011). IEEE, 2011; 1-8.
- [101] DE LUCIA A, QUSEF A. Requirements Engineering in Agile Software Development [J]. Journal of Emerging Technologies in Web Intelligence, 2010(2); 212-220.
- [102] BLOMKVIST J K, PERSSON J, ABERG J. Communication through Boundary Objects in Distributed Agile Teams [C] // Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM Press, New York, USA, 2015; 1875-1884.
- [103] BUCHAN J. An Empirical Cognitive Model of the Development of Shared Understanding of Requirements [C] // Communications in Computer and Information Science, 2014; 165-179.
- [104] RIVERO J M, GRIGERA J, ROSSI G, et al. Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering [M]. Information and Software Technology, 2014 (56); 670-687.
- [105] NAWROCKI J, OCHODEK M, JURKIEWICZ J, et al. Agile requirements engineering: A research perspective [J]. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, 17(3); 40-51.
- [106] ISSA A A, ALALI A I. Automated requirements engineering: use case patterns-driven approach [M]. IET Software, 2011(5); 287.
- [107] FARID W M. The NORMAP Methodology: Lightweight Engineering of Non-functional Requirements for Agile Processes [C] // 2012 19th Asia-Pacific Software Engineering Conference. IEEE, 2012; 322-325.
- [108] CHEN H, CHIANG R H L, STOREY V C. Business intelligence and analytics: from big data to big impact [R]. MIS Quart., 2012, 36(4); 1165-1188.
- [109] PACHECO C, GARCIA I, REYES M. Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques [M]. IET Software, 2018, 12(4); 365-378.
- [110] MANRIQUE-LOSADA B, ZAPATA-JARAMILLO C M, BURGOS D A. Re-expressing business processes information from corporate documents into controlled language [C] // International Conference on Applications of Natural Language Processing to Information Systems. Cham: Springer, 2016; 376-383.
- [111] HAUKSDOTTIR D, RITSING B, ANDERSEN J C. Establishing reusable requirements derived from laws and regulations for medical device development [C] // 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW). 2016; 220-228.
- [112] KAIYA H, SAEKI M. Using domain ontology as domain knowledge for requirements elicitation [C] // 14th IEEE International Requirements Engineering Conference (RE'06). 2006; 189-198.
- [113] LI Z Y, WANG Z X, YANG Y Y, et al. Towards a multiple ontology framework for requirements elicitation and reuse [C] // 31st Annual International Computer Software and Applications Conference (COMPSAC 2007). IEEE, 2007; 189-195.
- [114] NOGUEIRA F A, DE OLIVEIRA H C. Application of heuristics in business process models to support software requirements specification [C] // ICEIS. 2017; 40-51.
- [115] BENDJENNA H, ZAROOUR N E, CHARREL P, MAMIE A. A methodology to elicit requirements in inter-company co-operative information systems [C] // 2008 International Conference on Computational Intelligence for Modelling Control & Automation, 2008.
- [116] SHAO F, PENG R, LAI H, et al. DRank: a semi-automated requirements prioritization method based on preferences and dependencies [J]. Journal of Systems and Software, 2017, 126; 141-56.
- [117] ABAD Z, KARRAS O, GHAZI P, et al. What works better? A study of classifying requirements [C] // 2017 IEEE 25th International Requirements Engineering Conference (RE). IEEE, 2017; 496-501.
- [118] HAYES J H, ANTONIOL G, ADAMS B, et al. Inherent characteristics of traceability artifacts: Less is more [C] // 2015 IEEE 23rd International Requirements Engineering Conference (RE). IEEE, 2015; 196-201.
- [119] KAMALRUDIN M, HOSKING J, GRUNDY J. MaramaAIC: tool support for consistency management and validation of requirements [J]. Automated Software Engineering, 2017, 24(1); 1-45.

- [120] AHMED M A, BUTT W H, AHSAN I, et al. A novel natural language processing (NLP) approach to automatically generate conceptual class model from initial software requirements[C]// International Conference on Information Science and Applications, Singapore: Springer, 2017: 476-484.
- [121] KIFETEW F, MUNANTE D, PERINI A, et al. DMGame: A gamified collaborative requirements prioritisation tool[C]// 2017 IEEE 25th International Requirements Engineering Conference (RE). 2017: 468-469.
- [122] AHMAD S, JALIL I, AHMAD S. An enhancement of software requirements negotiation with rule-based reasoning: a conceptual model[J]. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 2016, 8(10): 193-198.
- [123] GROEN E C, IESE F, SCHOWALTER J, et al. Is there really a need for using NLP to elicit requirements? A benchmarking study to assess scalability of manual analysis[R]. REFSQ Workshops, 2018.
- [124] GROEN E C, SEYFF N, ALIR, et al. The crowd in requirements engineering: the landscape and challenges[J]. IEEE Software, 2017, 34(2): 44-52.
- [125] FERGUSON M. Big data-why transaction data is mission critical to success[J]. England: Intelligent Business Strategies, 2014, 2(1): 2-12.
- [126] MAALEJ W, NAYEBI M, JOHANNT, et al. Toward data-driven requirements engineering[J]. IEEE Software, 2016, 33(1): 48-54.
- [127] GENC-NAYEBI N, ABRAN A. A systematic literature review: opinion mining studies from mobile app store user reviews[J]. Journal of Systems Software, 2017, 125: 207-219.
- [128] MARTIN W, SARRO F, JIA Y, et al. A survey of app store analysis for software engineering[J]. IEEE Transactions on Software Engineering, 2017, 43(9): 817-847.
- [129] TAVAKOLI M, ZHAO L, HEYDARI A, et al. Extracting useful software development information from mobile application reviews: a survey of intelligent mining techniques and tools[J]. Expert Systems with Applications, 2018, 113: 186-199.
- [130] GHASEMI M, AMYOT D. From event logs to goals: a systematic literature review of goal-oriented process mining [J]. Requirements Engineering, 2019, 25(1): 67-93.
- [131] ZOWGHI D, COULIN C. Requirements elicitation: A survey of technique, approaches and tools[J]. Engineering and managing software requirements, 2005: 19-46.
- [132] ARRUDA D, MADHAVJI N H. State of requirements engineering research in the context of Big Data applications[C]// International Working Conference on Requirements Engineering: Foundation for Software Quality. Cham: Springer, 2018: 307-323.
- [133] DHINAKARAN V T, PULLE R, AJMERI N, et al. Appreview analysis via active learning: Reducing supervision effort without compromising classification accuracy[C]// 2018 IEEE 26th International Requirements Engineering Conference (RE). IEEE, 2018: 170-181.
- [134] DEOCADEZ R, HARRISON R, RODRIGUEZ D. Automatically classifying requirements from app stores: A preliminary study [C]// 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). 2017: 367-371.
- [135] GUZMAN E, ALKADHI R, SEYFF N. A needle in a haystack: What do twitter users say about software[C]// 2016 IEEE 24th International Requirements Engineering Conference (RE). IEEE, 2016: 96-105.
- [136] LU M, LIANG P. Automatic classification of non-functional requirements from augmented app user reviews[C]// Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, 2017: 344-353.
- [137] KUEHL N. Needmining: towards analytical support for service design[C]// International Conference on Exploring Services Science. Cham: Springer, 2016: 187-200.
- [138] MORALES-RAMIREZ I, KIFETEW F M, PERINI A. Speechacts based analysis for requirements discovery from online discussions[J]. Information Systems, 2018, 86: 94-112.
- [139] ZHAO L, ZHAO A. Sentiment analysis based requirement evolution prediction[J]. Future Internet, 2019; 11(2): 52.
- [140] SUN D, PENG R. A scenario model aggregation approach for mobile app requirements evolution based on user comments. Requirements engineering in the big data era[J]. Berlin: Springer, 2015: 75-91.
- [141] JIANG W, RUAN H, ZHANG L. Analysis of economic impact of online reviews: an approach for market-driven requirements evolution[J]. Berlin: Springer, 2014, 432: 45-59.
- [142] GUZMAN E, LBRAHIM M, GLINZ M. A little bird told me: Mining tweets for requirements and software evolution[C]// 2017 IEEE 25th International Requirements Engineering Conference (RE). IEEE, 2017: 11-20.
- [143] GUZMAN E, ALKADHI R, SEYFF N. An exploratory study of twitter messages about software applications[J]. Requirement Engineering, 2017, 22(3): 387-412.
- [144] CLELAND-HUANG J, DUMITRU H, DUAN C, et al. Automated support for managing feature requests in open forums [C]// Communications ACM, 2009: 68-74.
- [145] LIANG W, QIAN W, WU Y, et al. Mining context-aware user requirements from crowd contributed mobile data[C]// Proceedings of the 7th Asia-Pacific Symposium on Internetware. 2015: 132-140.
- [146] VOET H, ALTENHOF M, ELLERICH M, et al. A framework for the capture and analysis of product usage data for continuous product improvement[J]. Journal of Manufacturing Science and Engineering, 2019; 141(2): 021010.
- [147] BARNAGHI P, WANG W, HENSON C, et al. Semantics for the internet of things: early progress and back to the future[C]// International Journal on Semantic Web and Information Systems. 2012: 1-21.
- [148] NUSEIBEH B, EASTERBROOK S. Requirements engineering: a roadmap[C]// Proceedings of the Conference on the Future of Software Engineering. ACM, New York, 2000: 35-46.
- [149] DAMIAN D, CHISAN J. An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality, and risk management[J]. IEEE Transactions on Software Engineering, 2006, 32(7): 433-453.
- [150] MÉNDEZ FERNANDEZ D, WAGNER S, LOCHMANN K,

- et al. Field study on requirements engineering: investigation of artifacts, project parameters, and execution strategies[J]. *Information Software Technology*, 2012, 54(2): 162-178.
- [151] EVELEENS J, VERHOEF T. The rise and fall of the chaos report figures[J]. *IEEE Software*, 2010, 27(1): 30-36.
- [152] BUSCHERMOHLE R, EEKHOFF H, JOSKO B. success-Erfolgs- und Misserfolgsk Faktoren bei der Durchführung von Hard- und Software entwicklungsprojekten in Deutschland[M]. 2006.
- [153] KAMATA M, TAMAI T. How does requirements quality relate to project success or failure? [C]// *Proceedings of the 15th International Conference on Requirements Engineering*. IEEE Computer Society, 2007: 69-78.
- [154] NIKULA U, SAJANIEMI J, KILVIAINEN H. A state-of-the-practice survey on requirements engineering in small and medium-sized enterprises. *Research Report (951-764-431-0)* [R]. Telecom Business Research Center Lappeenranta, 2000.
- [155] STAPLES M, NIAZI M, JEFFERY R, et al. An exploratory study of why organizations do not adopt CMMI[J]. *Journal of Systems & Software*, 2007, 80(6): 883-895.
- [156] NAPIER N, MATHIASSEN L, JOHNSON R. Combining perceptions and prescriptions in requirements engineering process assessment: an industrial case study[J]. *IEEE Transactions on Software Engineering*, 2009, 35(5): 593-606.
- [157] PETERSSON F, IVARSSON M, GORSCHKEKT, et al. practitioner's guide to light weight software process assessment and improvement planning [J]. *Journal of Systems & Software*, 2008, 81(6): 972-995.
- [158] BEECHAM S, HALL T, RAINER A. Software process improvement problems in twelve software companies: an empirical analysis[J]. *Empirical Software Engineering*, 2003, 8(1): 7-42.
- [159] HSIA P, DAVIS A, KUNG D. Status report: requirements engineering[J]. *IEEE Software*, 1993, 10(6): 75-79.
- [160] KANG Y, LI H, LU C, et al. A transfer learning algorithm for automatic requirement model generation[J]. *Journal of Intelligent & Fuzzy Systems*, 2019, 36(2): 1183-1191.
- [161] MARTENS D, MAALE J W. Extracting and analyzing context information in user-support conversations on twitter[C]// *2019 IEEE 27th International Requirements Engineering Conference (RE)*. IEEE, 2019: 131-141.
- [162] SOLEMON B, SAHIBUDDIN S, GHANI A. Requirements engineering problems and practices in software companies: an industrial survey. *Advances in software engineering*[M]. Berlin: Springer, 2009: 70-77.
- [163] LIU L, LI T, PENG F. Why requirements engineering fails: a survey report from china[C]// *18th IEEE International Requirements Engineering Conference (RE)*. IEEE, 2010: 317-322.
- [164] VERNER J, COX K, BLEISTEIN S, et al. Requirements engineering and software project success: an industrial survey in Australia and the us[J]. *Australasian Journal of information systems*, 2007, 13(1): 225-238.
- [165] ALZAYED A. Requirements Engineering: An Industrial Survey In The Gulf Cooperation Council Countries [J]. *International Journal of Software Engineering (IJSE)*, 2021, 9(1): 107-129.
- [166] MÉNDEZ FERNÁNDEZ D. Naming the pain in requirements engineering Contemporary problems, causes, and effects in practice. *Springer Science+Business Media New York*[J]. *Empirical Software Engineering*, 2016, 22(5): 2298-2338.



WANG Hao-yu, born in 1997, master. His main research interests include requirements engineering and so on.