

# 导致软件开发外包失败的工程需求问题

兰德尔·福特, 戴维斯·帕森斯, 阿米蒂·卡舍, 戈特·巴恩

University of Greenwood Engineering, 4821 N Pennsylvania Ave, Independence, SK 30167, United States

**摘要:** 软件开发外包 (SDO) 因其降低成本、流程增强和应对所需资源稀缺等优势而越来越为人所知。不幸的是, 大多数软件开发外包项目未能实现预期收益。对此类项目失败的调查表明, 在某些情况下, 软件开发外包项目由于与需求工程过程相关的问题而失败。本研究的目的是识别和排序软件开发外包背景下需求工程 (RE) 过程中的常见问题。为此, 对当代文献进行了批判性评估, 确定了从业者面临的问题, 并由经验丰富的软件开发外包从业者组织了三份问卷。还使用了德尔菲技术、截止方法和 50% 规则。本研究确定了 SDO 案例中 RE 过程问题的类别。这些潜在类别是: 沟通, 知识管理和意识, 文化多样性, 信任, 诉管理和协调, 组织结构, 流程和工具, 利益相关者之间的关系, 以及以需求为中心。基于对 SDO 行业从业者的问卷调查, 应用 50% 规则, 除信任和组织结构之外的 7 个类别被选为 SDO 案例中 RE 流程问题的类别。至少 50% 或更多的调查参与者选择这七个类别作为 SDO 案例中 RE 流程问题的类别。

**关键词:** 需求工程, 软件工程师, 职业道德, 流行病学模型, 软件工程标准

## Requirements engineering issues causing software development outsourcing failure

Randell Ford, Davis Parsons, Amity Kasher, Gotter Barn

University of Greenwood Engineering, 4821 N Pennsylvania Ave, Independence, SK 30167, United States

**Abstract:** Software development outsourcing (SDO) is becoming more and more famous because of the advantages like cost abatement, process enhancement, and coping with the scarcity of needed resources. Studies confirm that unfortunately a large proportion of the software development outsourcing projects fails to realize anticipated benefits. Investigations into the failures of such projects divulge that in several cases software development outsourcing projects are failed because of the issues that are associated with requirements engineering (RE) process. The objective of this study is the identification and the ranking of the commonly occurring issues of the requirements engineering process in the case of software development outsourcing. For this purpose, contemporary literature has been assessed rigorously, issues faced by practitioners have been identified and three questionnaire surveys have been organized by involving experienced software development outsourcing practitioners. This study identifies categories of RE process problems in SDO cases. These potential categories are: Communication, Knowledge Management and Awareness, Cultural Diversity, Trust, V. Management and Coordination, Organizational Structure, Processes and Tools, Relationships between Stakeholders, and Needs-Centricity. Based on a questionnaire survey of SDO industry practitioners, applying the 50% rule, 7 categories other than trust and organizational structure were selected as categories for RE process issues in SDO cases. At least 50% or more of survey participants selected these seven categories as categories for RE process questions in SDO cases.

**Keywords:** requirement engineering, software engineer, professional ethics, epidemiological models, standards for software engineering

## 引言:

在信息技术外包期间,部分或全部与IT相关的功能被外包给外部供应商。一类信息技术外包是软件开发外包(SDO),它涉及将部分或全部与软件开发相关的任务外包给供应商。SDO的概念正在迅速普及,因为它促进了双方的利益。欧洲公司将软件开发外包给印度、越南和中国等国家。

外包有两个主要原因:一、外包的优势,如节省成本、利用优势技术和能力、优化内部资源利用;二、组织限制,例如管理不善和适当资源的稀缺。供应商受益于丰富的专业知识和对如何满足客户要求的理解。因此,供应商能够为其客户的供应链增加显著的价值。有几种类型的SDO,例如 onshoring、nearshoring、offshoring、分布式软件开发和全球软件开发(GSD)。这些项目被外包用于软件开发以获得预期的优势,但一些危险与SDO相关。此类项目的失败率很高,例如40%的海上项目没有达到预期收益。GSD有50%的故障率。调查显示,SDO的成功率只有50%。源自需求工程(RE)过程的问题是SDO失败的主要原因之一。

RE是软件开发生命周期(SDLC)中最关键的活动,对其他SDLC活动也有重大影响。一项研究表明,在SDLC期间经常发生与RE相关的错误。根据对12家软件开发公司面临的RE问题的行业调查,与RE相关的错误占有SDLC错误的48%。由于利益相关者的物理分散,这些问题在SDO的背景下以多种方式放大。因此,在SDO的情况下,RE过程会产生很多问题。因此,有必要识别和优先考虑在SDO的RE过程中经常发生或出现的问题,以设计积极的策略来解决SDO的失败,从而获得SDO的好处。在寻找SDO的RE流程的常见问题时,您还应该了解此类问题的类别,以便您可以将问题分类到适当的类别中。

## 工程与软件工程的定义与挑战路程

软件工程是一门年轻的子学科,力求属于工程专业。为了充分理解软件开发过程之外的挑战,有必要在更广泛的工程专业背景下重新审视软件工程。工程师专业发展委员会(ECPD)将工程领域定义为“科学原理的创造性应用……所有这些都尊重预期功能、运营经济性以及生命和财产安全。”基本上,工程师是创造性地将科学原理应用于产品设计(即结构、发动机、机器、设备、制造过程等)以实现特定功能的专家。因此,工程师非常熟悉“黑盒”或正在开发的软件产品的来龙去脉,以及如何使该产品适应软件应用程序的每个特定环境。此外,

与技术人员相比,工程师有望以创新和创造性的方式进行创新,以应对新技术的挑战。

由于软件工程是工程领域的一个相对较新的人,关于它是否符合工程的经典定义仍然存在很多争论。要解决这个问题,我们必须回答这个问题:什么是软件工程?这个问题首先在1968年的北约公约中得到解决,其中软件工程被定义为“建立和使用合理的工程原理以获得在真实机器上可靠和有效的经济软件”。1993年,计算机协会(ACM)和电气和电子工程师学会计算机学会(IEEE-CS)这两个世界领先的计算机学会成立了IEEE-CS/ACM软件建立联合指导委员会工程作为一种职业。

## 工程师与软件工程师应有的专业精神

关于职业定义的学术辩论,以及试图确定职业必须满足的特定标准才能被视为职业,是一个广泛的问题。一些人认为,几乎任何类型的职业(或实践)中的任何专家(基于知识的从业者)都可以被理解为专业实践的一部分。同样,Parsons(1939)将专业精神描述为现代性力量的自然伴奏,因此它的存在使从业者和公众都受益。相比之下,狭义的专业主义观点认为,并非所有职业都是职业职业。根据这种狭义的范式,职业是具有某些特征或特征的职业的专门子集。因此,职业必须满足某些要求才能被视为专业职业。

也许最早为职业设定标准的尝试是亚伯拉罕·弗莱克斯纳(Abraham Flexner)1915年臭名昭著的演讲“社会工作是一种职业吗?”他在其中宣称:他们从科学和学习中获取原材料;他们处理的材料具有实际和明确的目的;他们有教育传播的技术;他们倾向于自我组织;他们的动机越来越无私。Ernest Greenwood(1957)引用了该专业的五个属性:系统理论、权威、社区接受度、伦理和文化。Bayles(1982)确定了专业实践的三个必要特征:广泛的培训、涉及重要的智力成分,以及使一个人能够为社会提供重要的服务。在职业道德领域,工程职业道德是一个重要的子领域。在本文的这一部分,我们将首先概述工程师的职业道德。

在一项1996年论文的结论中,Michael Davis(1996)对软件工程师采取了非常关键的立场。他问了两个关于软件工程师的问题:第一,作为一个职业,“软件工程”真的是工程吗?其次,它在道德上是一种职业吗?最后,他对这两个问题的回答都是否定的。正如他所说,“没有迹象表明专业工程正在发生”。戴维斯的专业标准(1999)是规范性的而不是本体论的。从规范的角度来

看,他认为软件工程师的活动既不构成工程也不构成道德专业职业:职业的历史是特定职业如何组织自身以使其成员遵守特定标准的历史,超出了法律、市场和道德的要求。职业的历史是组织、能力标准和行为标准的历史。

该声明表明,要使一个职业被承认为一种职业,仅仅作为一个行会来运作是不够的——“根据技术成就授予他们(在专业层面上)协会的成员资格”。戴维斯提出了两个更苛刻的要求:他认为职业生涯必须基于特定的知识,这在当今的软件工程中通常被认为是软件工程学位。他的另一项要求是本次讨论的基础,即承诺以某种方式使用这些知识,即按照某种行为准则以专业的方式使用这些知识。专业人士肩负着维护某种精神、超越规则的使命的特殊责任(Davis 1996)。这种“承诺根据某些准则使用知识并坚持某种精神”是通过使用道德准则来实现的。自戴维斯的文章以来,软件工程领域发生了很大变化。根据 Ford 和 Gibbs (1996) 的说法,2004 年版的《软件工程知识体系指南》贡献了表征工程专业的五个组成部分中的三个。首先是知识部分,包括专业教育和认证、认证或强制许可、技能发展和继续专业教育。另外两个组成部分是社区支持和通过专业协会(行会)制定的职业道德守则。

在讨论本文的核心部分——道德规范之前,我们先简单地看一下知识部分。在知识需求方面,当今世界既有 Java 等新颖的编程语言,也有 Eclipse 等强大的开发工具和平台。然而,代码开发或编码不再局限于一小群计算专业人员。事实上,“编码”的能力是全球性的,几乎无处不在。然而,随着编码技能随着人类知识和主动性的广度激增,以及对日益复杂和强大的软件产品的需求增长更快,软件创建社区面临着与软件创建过程相关的问题,其中原始编码只是一个相对简单的部分(萨默维尔 2004 年)。事实上,软件开发(过程)比单纯的编码要复杂得多。作为开发和开发软件系统所涉及的一组相关活动和过程,软件开发(过程)被定义为“正在设计的软件”(Pressman 2010),这表明在开发软件和相关技术、方法和方法方面需要更多的专业知识。工具开发。

软件工程是一个基于计算机科学的理论基础以及旨在开发软件系统的工程方法的实践专业知识来建立实践技能的职业。从谱系上讲,软件工程领域是从开发软件的计算机程序员/计算机科学家演变而来的,他们通常无法考虑系统的其他一些同样重要的(技术、人力和组织)方面。软件工程师和计算机科学家之间的区别在于,

当计算机科学家提出计算方法和解决方案时,软件工程师设计和实施将产生质量和经济效益的过程和方法。职业所需的另一个条件是存在职业道德守则。这种情况表明了道德在软件工程专业中的重要性,正如 SE2004 软件工程本科学位课程指南中所确定的那样:“毕业生需要理解和欣赏道德和职业行为、经济和社会需求”。

科学和工程领域的大多数道德规范在道德标准和主题上都非常相似。由 ACM 和 IEEE-CS 联合批准的软件工程师道德规范经过多年的重大修订,直到 5.2 版的产生,该版本于 1999 年被批准并接受为软件工程教学和实践的标准(Gotterbarn 1999)。当前的准则包括与关键利益相关者相关的八项原则:公众、客户和雇主、产品、判断、管理、专业、同事和自我。与所有道德决策一样,本准则的基本指导原则是对公众健康、安全和福利的首要关注。本守则的序言总结了以下内容:由于他们在开发软件系统中的作用,软件工程师有很大的机会做好事或造成伤害,促使他人做好事或伤害,或影响他人做好事或伤害。为了确保他们的努力得到尽可能充分的利用,软件工程师必须致力于使软件工程成为一个有价值 and 受人尊敬的职业。根据这一承诺,软件工程师应遵守以下道德规范和专业实践。

总结上面回顾的软件工程的定义,作为一个职业,它必须在质量、过程、方法和工具等方面遵守某些标准,以及其他问题,包括调度、经济、风险和编码。该准则的八项原则涉及工程师与以下利益相关者的关系:公众、客户和雇主、产品、判断、管理、专业、同事和自我。作为 Davis 的 Challenge (1999) 的后续,本文展示了软件工程师的道德框架如何适应任何软件的生命周期。通过弥合软件工程实践的道德和专业元素之间的差距,道德框架的实施最终将产生按照该行业的专业标准工作的工程师。根据定义,这些标准将涉及某些道德考虑。

### 了解软件工程的标准与提高其可靠性

与流行病学的所有方面一样,管理模拟开发的工程标准是规范性考虑与技术约束重叠的问题。因此,对这些模拟的严格审查并不是我们上面讨论的任何科学专家子集的专有特权。科学软件开发人员的实用指南在下面进行了描述和辩护。我们没有发明这些标准。相反,我们的提议借鉴了软件工程的历史。在过去的 5 年中,软件工程社区一直在努力编纂实践和程序,这些实践和程序已根据经验确定,以帮助最大限度地降低开发成本、进度和风险。这一努力导致了一系列不断发展的软件工程标准,其中最新的一个是 ISO 2017。我们注意到,科



学探究软件的开发通常不符合 ISO 2017 对安全关键软件的要求。因此, 我们的建议将是具有争议的, 并且可能会被那些只从术语中看待流行病学的人认为过于僵化的科学调查。

也许我们提案中最具争议的特点是将工程伦理方法应用于一门主要将自己视为科学研究的学科。虽然负责的工程实践通常对各种项目中涉及的危害很敏感, 但有人可能会争辩说, 像流行病学这样的科学是不同的。该论点认为, 科学研究伦理与工程伦理有很大不同。大多数科学哲学家可能会同意。例如, 流行病学家产生的模拟类型被哲学家视为正式或抽象的对象, 或者是能够产生关于他们所模拟的系统的经验信息的特殊形式的实验。然而, 这种对模拟器 (以及更普遍的软件) 的看法忽略了与模拟器的可靠性和可信度直接相关的广泛的一般“工程”问题。例如, 关于我们如何帮助确保模拟器中的组件开发人员清楚地了解软件是如何开发并与其他人编写的软件集成的问题, 以及哪些文档化的编程和验证实践有助于最大限度地提高可靠性如果我们考虑这些问题, 模拟器软件不会以抽象的对象或特殊的实验形式出现。继 Alvarado (2020) 之后, 我们认为流行病学模拟除了是正式的模型或实验外, 还应被理解为工程科学工具。一般来说, 科学仪器应该满足“故障安全”的工程开发标准, 以解决刚才提到的工程问题。

正如 Roddis (1993) 所指出的, 在工程伦理中, 管理工具和实践的标准至少部分取决于失败可能造成的伤害。我们认为, 鉴于在公共政策决策中部署这些工具所涉及的高失败成本, 流行病学模拟器中的软件工程需要高风险管理标准。我们认为, 在可能造成巨大伤害的情况下, 科学家、资助机构和政府应采用至少与社会在关键基础设施、航空或军事环境中通常要求的一样高的软件工程标准。由于这些模拟中的错误可能对社会造成的危害的性质和程度, 流行病学建模者有特殊的义务在软件开发中遵守高标准。

对遵守此类标准的一个反对意见是, 趋同于一套单一的标准可能会抑制或减缓科学研究的发展。我们认为, 从长远来看, 这种风险并不大, 按照高风险管理标准构建的开放透明的科学软件可能有助于而不是阻碍科学事业。像 ISO 2017 这样的标准, 凭借其包含的许可条款, 是高度适应特定的风险制度。例如, 这些标准肯定会允许以不必满足“故障安全”标准的方式开发某些模拟器 (例如仅用于辅助查询的模拟器)。ISO 2017 要求其他模拟器, 例如用于验证核反应堆安全的模拟器, 必须按照

“极度偏执”的安全标准建造。这里的关键是, 工程标准是有意识地通过对相关系统开发所涉及的风险的判断来塑造的。参与公共政策决策的流行病学模拟器显然涉及极高的伤害风险。

## 结论

考虑到软件开发外包 (SDO) 的预期收益及其失败的原因, 本研究探讨并强调了 SDO 环境中需求工程 (RE) 过程中的常见问题。很多时候, RE 流程问题会危及 SDO 项目并最终导致此类项目失败。为了规避解决 SDO RE 过程问题的“救火”方法, 并成功解决此类问题以获得 SDO 收益, 必须提前从“发生频率”的角度考虑问题。本研究探讨了 SDO 的 RE 过程的问题。这些问题分为不同的类别。因此, 本研究首先确定了 SDO 的七类 RE 流程问题, 它们是: 一、沟通, 二、知识管理和意识, 三、文化多样性, 四、管理和协调, 五、流程和工具, 六、利益相关者之间的关系, 以及 七、以需求为中心 (RQ1)。

本研究还解释了 SDO RE 流程的 10 大问题。识别常见的 SDO RE 流程问题并确定其优先级有助于高管和经理计划积极的战略来解决 SDO RE 流程问题, 从而实现 SDO 的预期收益。作为未来的工作, 计划主要是在 SDO 的情况下, 确定 RE 过程中常见问题的根本原因, 并对其进行根本原因分析。其次则是旨在解决 SDO RE 流程问题的模型。这些项目被外包用于软件开发以获得预期的优势, 但一些危险与 SDO 相关。此类项目的失败率很高, 例如 40% 的海上项目没有达到预期收益。GSD 有 50% 的故障率。调查显示, SDO 的成功率只有 50%。源自需求工程 (RE) 流程的问题是 SDO 失败的主要原因之一。

本文献中的几项研究集中在 SDO RE 过程问题上。在这项研究中, 主要关注的是 GSD 中的“需求理解”。因此, 分布式软件开发强调对 RE 相关活动的透彻理解, 这需要分散的利益相关者的集体努力。已经提出并测试了一个称为 PBURC 的框架, 用于在涉及不同上下文和服务的 RE 过程中收集和验证数据。已经描述了使用 MAS (多代理系统) 架构来减少分布式 RE 过程的问题, 特别是对于验证和确认。为了了解 GSD RE 过程的卷积, 在需求定义过程中分析了 24 个虚拟团队的能力。通过实地研究, D. Damian 调查了地理分布的利益相关者对可再生能源过程的影响。基于 RE 相关任务和 GSD 问题的暴露, 已经提出并评估了几个 GSD RE 模型。在利益相关者分散的情况下, 建议使用 V 模型来提取和选择产品发布的需求。全球可再生能源案例中的知识分配和再利用已经

在争论中。为了解决大量分布式终端用户的挑战，引入统一在线方法。

**文献参考:**

[1]M. Niazi, M. El-Attar, M. Usman, and N. Ikram. (2012, May). “GlobReq: A framework for improving requirements engineering in global software development projects: Preliminary results,” in Proc. 16th International IET Conference on Evaluation & Assessment in Software Engineering, pp. 166 - 170, 2012.

[2]Sayão M., Haendchen A. F., and do Prado H. A. “Requirements engineering for distributed development using software agents,” *Advances in Conceptual Modeling - Challenges and Opportunities*, pp. 272 - 281, 2008.

[3]Okoli C., and Pawlowski S. D. “The Delphi method as a research tool: an example, design considerations and applications,” *Information & management*, vol. 42, no. 1, pp. 15 - 29, 2004.

[4]Vizcaíno A., García F., Villar J. C., Piattini M., and Portillo J. “Applying Q-methodology to analyse the success factors in GSD,” *Information and Software Technology*, vol. 55, no. 7, pp. 1200 - 1211, 2013.

[5]Amity, E. (2014). *Agile and professional ethics*. M.Sc.

Thesis in software engineering. SCE – Shamoon College of engineering, Israel.

[6]Gaumnitz, B. R. (2004). A classification scheme for codes of business ethics. *Journal of Business Ethics*, 49(4), 329 - 335.

[7]NIST. (2002). New Release of June 28, 2002. The National Institute of Standards and Technology. Parsons, T. (1939). *The professions and social structure*. In T. Parsons (Ed.), *Essays in sociological theory* (pp. 34 - 49). New York: The Free Press.

[8]Keller, E. F. (2002). *Making sense of life: Explaining biological development with models, metaphors and machines*. Cambridge, MA: Harvard University Press.

[9]Sharma, D. (2020). What does it really mean to ‘follow the science’? *The Pharmaceutical Journal*, PJ June 2020 online, online | DOI: <https://doi.org/10.1211/PJ.2020.20208008>. Accessed August 10, 2020.

[10]Winsberg, E. (2019). Computer simulations in science. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (summer 2015 edition). <https://plato.stanford.edu/archives/sum2015/entries/simulations-science/>. Accessed 20 Dec 2018.