

软件需求中的非规范信息

牟克典

摘要 对绝大多数软件开发项目来说，获取完全准确的需求信息并非能够一蹴而就的。在确定软件需求过程中不得不面对各种非规范信息。如果对非规范信息进行有效管理，不仅可以改进确定需求过程、提高软件质量，而且能有效控制开发成本。本文主要介绍了需求阶段常见的非规范信息以及相应的管理方法、策略等方面的研究进展。

1 引言

软件需求规格说明作为待开发系统的描述基线，为设计、编码、测试、维护以及演化等软件开发的其它阶段提供基础。软件需求规格说明的质量对软件开发项目的成败具有重要影响。有关研究表明绝大多数软件开发项目的失败都可归结到不合格的需求信息，软件产品中的大多数错误来源于需求确定阶段^[1,2]。欧洲改进软件开发进程训练计划— European Software Process Improvement Training Initiative (ESPITI) 在 1995 通过调查公布了软件开发过程中各个阶段的错误和缺陷统计情况，如下图 1 所表示^[2]。

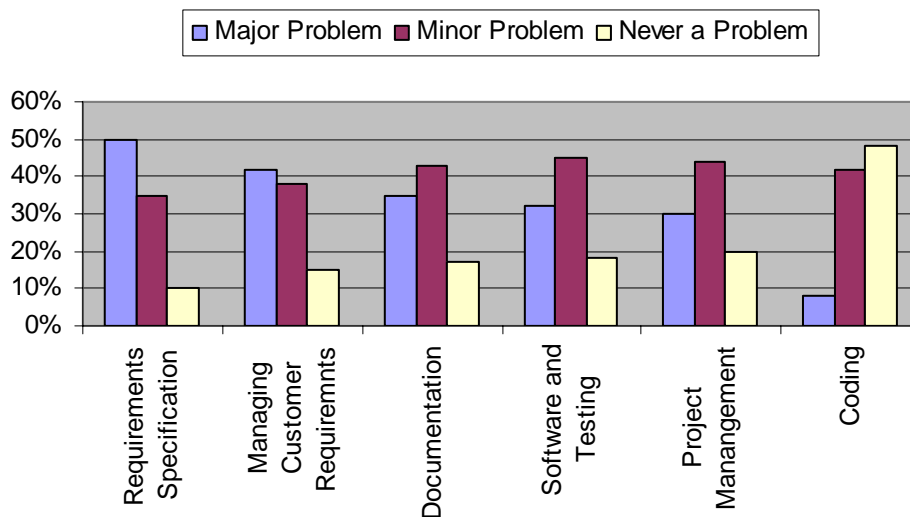


图 1 软件开发中各阶段的错误和缺陷分布情况

另一方面，由不恰当的需求信息所导致的错误如果在软件测试甚至运行阶段才被检测到，修复这些错误的代价是十分高昂的^[3,4,5,6]。Davis曾经给出了在软件开发的的不同阶段修复同一错误的相对代价^[2,6]，如下图 2 所示。

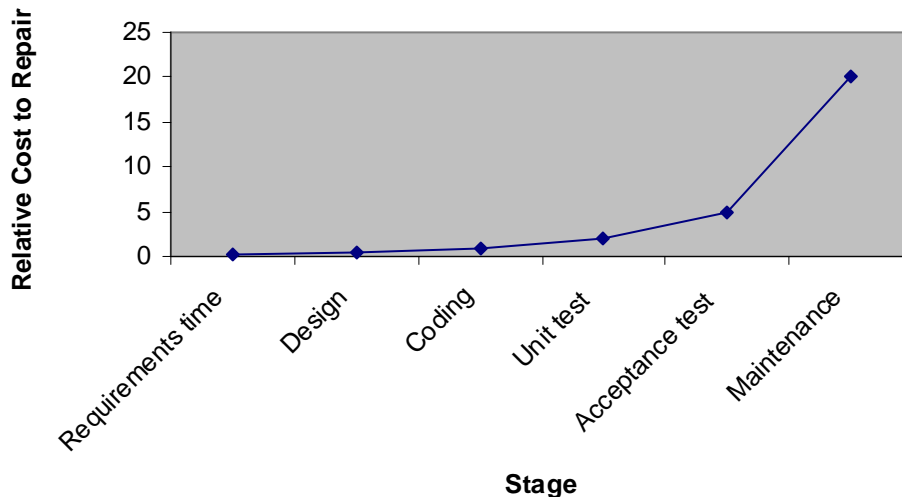


图 2 在软件开发的阶段修复错误的相对代价

从这个意义上讲，在需求确定过程中完全根除或者尽可能减少需求错误的个数对降低软件测试甚至运行阶段的高昂维修费用具有关键作用^[4,5]。获取优秀的软件需求规格说明是提高未来软件系统质量、控制开发成本、节约维修费用的有效手段。

尽管目前没有统一的形式化标准来区分“优秀”的需求描述和“不合格”的需求描述，但是通常认为“优秀”的单个需求描述不仅是完整的、正确的、必要的、无歧义的，而且是可行的、可验证的以及被设置了优先级别的^[5,6,7]。对于“优秀”的软件需求规格说明而言，通常认为完整性、一致性、可跟踪性和可修改性是其应该具备的基本特征^[5]。

事实上，在软件开发实践中获取完整、协调、明晰、准确的需求规格说明并非易事，在需求抽取阶段不可避免存在如下困难和障碍^[8]：

- 在软件需求获取的初期阶段，需求分析人员收集到的需求信息经常是零碎的、不系统的，甚至是模糊的；
- 客户或者应用领域的专家都不习惯表达他们认为是常识的知识或信息，使得提供给软件开发人员的需求信息可能是不完整的；
- 在很多情况下，软件开发人员对应用领域并不熟悉，只能基于客户和领域专家提供的信息来抽取需求；而客户和领域专家经常不知道如何以软件开发人员熟悉的术语和方式表达他们的需求信息。他们之间的通讯鸿沟（Communication Gap）导致他们之间的相互误解；
- 在多视点^[9]、多风险承担者参与的需求获取模式中，各风险承担者不同的技术背景、职责、利益立场等不可避免地导致需求获取中的分歧、矛盾以及冗余的信息等^[10,11]。

尽管有些当前的需求工程方法和技术可以解决或者缓解某些困难（例如基于本体的领域建模方法^[8]可以有效处理通讯鸿沟问题），但没有一种有效的方法或技术可以完全根除这些常见的困难和障碍。因此，在软件开发实践中我们经常不得不面对如下问题：

- 模糊的需求信息；
- 不完整的需求信息；

- 冗余的需求信息；
- 不一致的需求信息；
- 有分歧的质量属性要求；
- 需求优先级设置上的冲突^[12]等等。

在文^[13]中，这些有问题的需求信息被统称为“非规范需求”信息。这些非规范需求信息对软件工程的其它后继阶段和软件质量都有直接影响，例如：

- 模糊的或者有分歧的需求信息增加了软件工程设计、编码、测试等阶段的不确定因素；
- 不一致的需求信息既可能导致未来系统中的局部错误，也可能导致未来系统崩溃；
- 需求优先级设置上的冲突可能影响整个项目开发计划的安排等等。

尽管当前的大多数软件开发技术都力图回避这些非规范信息，但从软件的整个生命周期来看，非规范需求信息对软件开发过程的影响其实并不全是负面的。通常非规范信息是系统中由需求导致的错误的可能来源之一，对非规范需求信息进行有效管理，有助于在需求过程中尽可能减少需求错误、提高软件开发质量、改进软件开发过程。

2 非规范需求信息

对非规范需求的检测与修整是需求验证的主要任务，其中对不一致信息的管理是近来软件开发中特别关注的问题^[14,15,16,17,18]。在文^[13]中，以不一致信息为核心，将非规范信息狭义化，即用“非规范信息”表示与不一致信息相关的常见的有问题的需求信息，着重指冗余需求和模糊需求。其中冗余需求是需求描述集中不一致性的主要来源之一，而模糊需求是潜在不一致性的可能来源。不妨设 S 是基本的需求规格说明（或者部分规格说明）， D 表示与特定应用场景相关的事实， E 是 S 对应的未来系统在应用场景 D 中的期望运行结果，则不一致需求和冗余需求可以定义如下^[13]：

- **不一致需求**：检测 S 在应用场景 D 中是否存在不一致是指检测是否 $\exists \alpha \in E$ ，满足 $S \cup D \vdash \alpha \wedge \neg \alpha$ 。
- **冗余需求**：如果对于某个 $\emptyset \subset S_1 \subset S$ 来说，若 $S \cup D \vdash E$ 且 $(S - S_1) \cup D \vdash E$ ，或者 $S \cup D \vdash \alpha \wedge \neg \alpha$ 且 $(S - S_1) \cup D \vdash E$ 则我们可以认为 S_1 相对于应用场景 D 是可能的冗余需求集。

为了尽可能避免需求信息的遗漏，还需要考虑**模糊需求**^[13]。尽管不一致信息有助于软件开发人员发现比较模糊的、不确定的需求信息^[15]，但目前对不一致信息检测和分析的研究工作仅局限于考虑能明确表示的需求集，基本上忽略对模糊需求的考虑。其实模糊需求从本质上来说是一种“可能”的需求，基于当前的信息很难确切地表达出来，并且随着需求的精化，模糊需求也逐渐向可明确表示的需求转化。特别要注意到当这些模糊需求转变为可明确表示的需求以后，很可能导致新的不一致信息。在一般情况下，对模糊需求信息可以通过如下两种策略进行处理：

- **TBD (To Be Determined) 策略**^[5]：对于一些特别模糊或者分歧特别大的需求信息来说，很难在当前阶段做出相关的决定。处理这种情况最明智的策略就是暂时将这种信息标记为TBD，并建立可跟踪的TBD需求列表留待合适的开发阶段处理。TBD需求可能隐含着潜在的不一致信息。反过来，对不一致信息的进一步分析可能导致某些涉及的需求转变为需要进一步确定的TBD需求。
- **缺省策略**：此策略是软件开发中常用的表示变化性的策略。在通常情况下，对于可变化的属性来说，在软件开发过程中暂时推荐一个当前认为最可能的属性值作为缺省值。领域分析和建模是利用领域

知识来缓解需求获取困难、增强需求复用的有效途径之一^[8]。对于领域模型来说，一方面需要表示应用领域中类似系统之间的共同需求（共性），一方面又要具有一定的灵活性，能表示领域模型的变化性，以适应本领域不同的系统开发需要。在本文中我们采用缺省方式来表示领域模型中对应于变化性的需求。这样一方面表示这种需求没有最终确定，具有一定的不确定性，同时也表示基于当前的信息，我们可以暂时做出缺省决策。其实，即使没有领域模型，在软件开发过程中，为了避免遗漏需求信息，有经验的需求分析人员会以“一般情况下...”的方式向客户提出建议。如果客户不直接否决，则可以暂时以缺省策略处理这种建议。和TBD需求类似，我们可以建立可跟踪的缺省需求列表。如果缺省的需求信息和规范的需求信息相冲突，在很大程度上缺省的需求信息是被审查和修改的对象。

如果能对这些非规范需求进行有效管理，至少可以在以下几个方面改进软件开发过程、提高未来软件产品的质量、节约测试阶段的维修成本^[13,14,15]：

- 对模糊需求的考虑有利于避免遗漏潜在的需求信息；
- 采用缺省策略可以利用当前不完全的信息做出比较灵活的决策；
- 采用缺省策略有利于检测潜在的不一致信息；
- 利用不一致信息来改进软件开发过程；
- 通过检测冗余需求来减少不一致信息，并避免设计、编码、测试等阶段的开发资源浪费。

3 非规范需求信息管理框架

对非规范需求信息进行有效管理是以非规范信息为工具改进需求获取质量的关键。在非规范需求信息的管理方面，Nuseibeh等曾提出了一般意义上不一致性管理框架^[15]。文^[13]中对这种框架进行了拓展，提出了以容忍不一致性的需求推理为核心的非规范需求管理框架，如图3所示。

在此管理框架中，通过在特定的应用场景中对需求进行推理，并且借助于与特定应用场景相关的知识（例如设计的需求测试用例^[19]等）可以对推理结果进行分析：

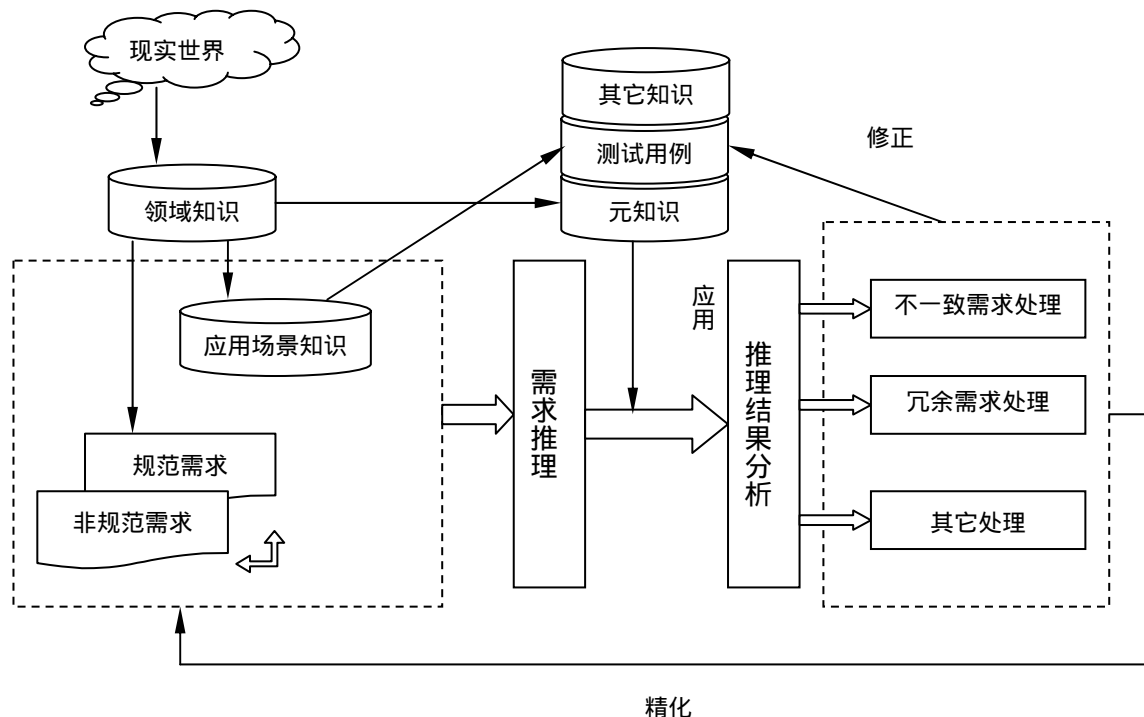


图3 非规范需求管理框架

- 如果结果中出现了不一致性，则可以采用 Nuseibeh 等提出的不一致性管理框架进行处理；
- 如果检测到冗余需求，则通过和客户的进一步协商来删除冗余信息；
- 如果没有得到期望的推理结果，则可能遗漏了某些信息，相关的开发人员和客户需要进一步的协作来确定是否遗漏信息，以及遗漏了那些信息。

通过对推理结果的分析 and 处理，可以进一步对基本需求规格说明和相关辅助知识采取一定的行动：

- 对基本的需求规格说明进行精化，包括规范需求和非规范需求的相互转化、补充新信息等；
- 对用于辅助推理结果分析的特定场景相关知识进行修正和完善。

需求形式化表示和推理工具显然是此非规范需求信息管理的核心问题，文献^[13,19]也详细论述了标记谓词演算（Annotated Predicate Calculus）是适合需求表示和推理的逻辑工具。

4 结论

大多数软件开发人员已经认识到有效管理非规范信息对提高软件产品质量、控制开发成本、降低维修成本方面具有重要意义。本文主要介绍了需求阶段常见非规范信息的种类以及当前关于非规范信息管理方面的研究进展。事实上，对于软件开发这样一个实践性很强的领域，本文所提到的方法或者管理框架虽然具有一定的指导作用，但距离软件开发实践还有相当的距离。如何针对特定的技术或者软件开发环境来具体管理非规范需求信息无疑是软件开发人员更为关心的问题，这也是值得研究人员重视的方向之一。

参考文献

- [1] Glass. R. Software Runaways [M], Harlow: Prentice Hall,1998,p21.
- [2] Leffingwell, and Don Widrig. 2003. Managing Software Requirements: A Use Case Approach. . Boston, MA: Addison-Wesley.
- [3] J. C. Kelly, J. S. Sherif, and J. Hops. An analysis of defect densities found during software inspections [J]. J. Syst. Softw. ,1992, 17(2):111–117.
- [4] P. Grunbacher, M. Halling, S., Biffl, An empirical study on groupware support for software inspection meetings [A], the Proceedings of 18th IEEE conference of ASE [C], Montreal Canada, 2003, 4-11.
- [5] K. E. Wiegers. Software Requirements [M]. Microsoft Press, 2003.
- [6] Davis, Alan, M., Software Requirements: Objects, Functions, and States [M], Englewood Cliffs, NJ:PTR Prentice Hall, 1993.
- [7] IEEE.1998, IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications [S]. Los Alamitos, CA: IEEE Computer Society Press, 1998.
- [8] Lu. R., Jin, Z. Domain Modeling-Based Software Engineering [M], Kluwer Publishers, 2001
- [9] A. Finkelsetin, J. Kramer, B. Nuseibeh, L. Finkelstein, M.Goedicke, Viewpoints: A Framework for Integrating Multiple Perspectives in System Development [J], International Journal of Software Engineering and Knowledge Engineering, 1992, 2(1), 31-58.
- [10] A. Silva, Requirements, Domain and Specifications: A Viewpoint-based Approach to Requirements Engineering [A], Proceedings of ICSE02 [C], Orlando, Florida, USA, May, 2002, 94-104
- [11] S. Easterbrook and M.Chechik. A framework for multivalued reasoning over inconsistent viewpoints [A]. In Proceedings of ICSE'01[C] , Toronto, Canada, May 2001, 411–420.
- [12] 牟克典,金芝,陆汝钤,视点合成中重叠需求的不一致优先级处理 [J],计算机学报, 2004, 27(10):1379-1387.
- [13] 牟克典 , 金芝 , 陆汝钤.非规范软件需求管理[J].《电子学报》,2004(12a).
- [14] B. Nuseibeh, S. Easterbrook, and A. Russo. Leveraging inconsistency in software development [J]. IEEE Computer, 2000,33(4):24–29.
- [15] B. Nuseibeh, S.Easterbrook, and A.Russo. Making inconsistency respectable in software development [J]. Journal of Systems and Software, 2001,58(2):171–180.
- [16] A. Finkelstein, D. Gabbay, A. Hunter, J. Kramer, and B. Nuseibeh. Inconsistency handling in multi-perspective specifications [J]. Transactions on Software Engineering, 1994, 20(8):569– 578.
- [17] G. Spanoudakis and A.Zisman. Inconsistency management in software engineering: Survey and open research issues [A]. In S.K.Chang, editor, Handbook of Software Engineering and Knowledge Engineering [C], World Scientific Publishing Co., 2001,329–380.
- [18] A. Hunter and B.Nuseibeh. Managing inconsistent specification [J]. ACM Transactions on Software Engineering and Methodology, 1998, 7(4):335–367.
- [19] Kedian Mu, Zhi Jin, Ruqian Lu, A logical Approach to testing requirements [A], to appear in Proceedings of IASTED conference on SE05.

作者：牟克典 中国科学院计算技术研究所智能信息处理重点实验室 博士后

