


计 算 机 科 学 丛 书

软件需求

Software
Requirements

(美) Karl E. Wiegerts 著

陆丽娜 王忠民 王志敏 等译

 机械工业出版社
China Machine Press

Microsoft Press

计算机科学丛书

软件需求

(美) Karl E. Wiegiers 著
陆丽娜 王忠民 王志敏 等译

 **机械工业出版社**
China Machine Press

本书讲述了软件开发中一个至关重要的问题——软件需求问题。软件开发人员及用户往往容易忽略信息沟通，导致软件开发出来后，不能很好地满足用户的需要。而返工则不仅在技术上给开发人员带来巨大的麻烦，而且软件性能深受影响且造成人力、物力的浪费。所以在开发周期早期提高项目需求分析的质量，减少重复劳动，通过控制项目范围的扩展及需求变更来达到按计划完成预定目标是当前我国软件业急需解决的问题——这也是本书讨论的主要内容。

Karl E. Wiegers: Software Requirements.

Copyright 2000 by Microsoft Corporation.

Original English language edition copyright © 1999 by Microsoft Corporation.

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A. All rights reserved.

本书中文简体字版由美国微软出版社授权机械工业出版社出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书版权登记号：图字：01-1999-3113

图书在版编目(CIP)数据

软件需求 / (美)威伊格斯(Wiegers, K. E.)著; 陆丽娜等译. —北京: 机械工业出版社, 2000.7

(计算机科学丛书)

书名原文: Software Requirements

ISBN 7-111-08127-7

I. 软… II. ①威… ②陆… III. 软件开发 IV.TP311.52

中国版本图书馆CIP数据核字(2000)第63908号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 周桦

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2000年7月第1版第1次印刷

787mm × 1092 mm 1/16 · 11.5印张

印数: 0 001-6 000册

定价: 19.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译者序

近年来在计算机领域中，软件工程的地位显得越来越重要，不仅大型项目离不开它，就是一般的甚至小型项目也离不开它。本书提供的许多工具将有助于信息交流，同时将帮助软件专业人员、管理者、市场营销者以及客户能应用有效的软件需求工程方法。本书还介绍了许多方法，用来帮助开发小组和客户共同理解怎样构造一个软件才能满足用户实际的需要，同时也包括了编写文档和管理变更的方法。本书的作者具有非常丰富的软件开发经验，他根据实际工作经验提炼的这本著作是计算机软件工程开发人员一本十分难得的好读物，为此，我们将它翻译出介绍给读者。

本书共分三部分。第一部分主要针对软件需求工程中的基本概念、主要步骤、典型特点、关键问题等作一个较为详细的介绍与说明。该部分分为5章，首先从探讨软件需求的本质入手，引出软件需求的概念定义及其重要性；第2章从用户角度强调了开发人员与用户之间的合作乃是成功的关键；第3章提供了软件需求中的一些方法以指导实践；第4章说明怎样改善需求过程；第5章对需求风险管理予以透彻的分析。

本书第二部分主要讲述了软件开发的全部过程，并给出许多文档实例以及各种分析方法和技巧。该部分分为10章，首先介绍了需求开发的四个步骤：需求的获取、分析、编写文档与评审，以及各步骤的任务和所用的方法。其次介绍了需求的图形化分析方法，用数据流程图、实体联系图、状态转换图、类图、对话图等方法建立模型以利于优化需求。最后介绍了原型法、设定需求优先级等常用的需求开发方法，以及从需求开发向设计规划转化时的一些方法和标准。

本书第三部分主要讲述软件需求管理的原则和策略，它分为4章。首先是建立一个需求变更控制，使用状态转换图、变更处理过程、问题跟踪工具来管理变更需求。接着要记录需求链中的联系，以利于在需求管理中快速定位及准确修改，按照影响分析的策略列出的步骤进行分析，可以正确、快速完成影响分析。最后介绍一些用于需求管理的工具和它的使用，及使用后带来的巨大好处。

本书第1~5章由王忠民副教授、王智翻译，第6~15章由陆丽娜教授、王志敏翻译，第16~19章由伍卫国、薛军翻译，全书由陆丽娜教授、王忠明副教授统稿。在本书翻译过程中，机械工业出版社华章公司的有关人员给予了极大的支持与关心，借此表示深深的感谢。由于译者水平有限，错误在所难免，请多多指教！

译者2000年6月8日
于西安交通大学

前 言

尽管拥有50年积累的经验，许多软件开发组织仍不得不在收集、编写和管理产品需求中疲于奔命。缺乏用户参与、不完整的需求及不断变更需求，是导致信息技术项目不能按进度安排和资金预算完成全部功能的主要原因(The CHAOS Report,The Standish Group International,Inc.,1995)。许多软件开发人员不能熟练地收集客户(customer)需求，很多开发者并不知道实用的需求工程技术，而且教学课程中也是技术主题比需求主题占有优势，工程参与者甚至对“需求”是什么也有不同的看法。

软件开发中，信息沟通(交流)至少应与计算占有同等的比重，然而我们往往强调了计算而忽略了信息沟通。本书提供的许多工具将有助于信息交流，同时将帮助软件专业人员、管理者、市场营销者以及客户应用有效的需求工程方法。本书还介绍了许多方法，用来帮助开发小组和客户理解怎样构造一个软件才能满足用户(user)实际的需要，同时也包括了编写文档和管理变更的方法。本书中介绍的技术代表需求工程中主流的“良好的习惯和做法”，而非来源于专业领域的高新技术或试图解决所有需求问题的复杂的方法学。

本书有益于读者之处

本书可帮助改进你着手的所有软件过程，对改善需求开发和管理实践都能提供很多的益处。本书是介绍概念和方法的，并不涉及专门的开发方法学或应用领域，所以它适用于各种项目。我尽力以清晰的结构风格介绍大量实用的且经过验证的技术，希望在以下几个方面能给你提供帮助：

- 使客户的满意度更高。
- 减少维护和支持费用。
- 在开发周期早期提高项目需求分析的质量，减少重复劳动，从而提高生产率。
- 通过控制项目范围的扩展(creep)及需求变更，按计划完成预定目标。

我的目标是帮助你改进收集和分析需求的过程、编写和验证需求规格说明以及在整个产品开发周期中管理需求。改进过程的最终目的是使你组织中的人员以新的方式进行工作，从而得到更好的结果。因此，希望你能将所学用于实践，而不仅是“纸上谈兵”。

实例研究

为帮助读者理解怎样应用本书介绍的各种方法，书中提供了几个基于实际项目的实例，其中一个中等规模的信息系统——“化学制品跟踪系统”说明了许多实用技术(不必担忧——你无需知道任何关于化学的知识也能理解该项目)。这个实例的项目说明还会帮助你了解怎样把不同的策略(方法)较好地融合到一块。本书中还穿插着各实例研究项目参与者之间的对话实例。不论你的小组是开发什么软件的，这些对话总是对你有益的。

谁应该读这本书

凡参与一个新的或升级的软件产品的需求定义或说明中的任何人员都能从本书中获得有

用的信息。这些人中包括那些必须理解并满足用户期望的分析、开发、测试人员；也包括用户，这些用户想定义一种符合他们功能和使用需要的产品。希望确认产品是否满足业务需要的客户将能更好地理解需求分析过程的重要性。而负责监督按期完成产品的项目管理者也将学会怎样管理潜在的威胁——需求变更。

在许多培训班的课程中，我发现那些非技术方面的项目参与者也很容易理解本书所涉及的内容。想提高自己对开发过程的理解并想了解需求在软件成功中扮演的关键角色的任何人都将从本书中受益。

本书结构

本书分为三大部分。第一部分先介绍了一些基本的需求工程定义和一些优秀的需求分析所具有的特性。我希望你与你的重要客户一起阅读第2章（关于客户与开发者之间的伙伴关系）；第3章介绍了许多在业界采用的需求开发和管理的“好方法”（良好的习惯做法）。第4章有助于计划怎样将新的实践融入小组的开发过程中。方法是基于对附录中当前需求实践自我测试的回答。第5章则介绍了一些通常与需求相关的项目风险。

第二部分介绍了许多关于需求开发的技术。首先是定义项目的业务需求，项目视图(vision)及涉及的范围(scope)。接下来介绍怎样为项目寻找合适的客户代表，获取(elicit)客户需求，编写功能需求文档及质量属性文档。第10章介绍了一些分析模型，这些模型可用于不同范围的需求分析。第12章介绍了软件原型的结构和应用。第二部分还探讨了定义需求优先级的方法及验证需求分析是否正确的方法。

第三部分的主题是需求管理的原则和实施办法。这部分还特别介绍了处理需求变更和评价每项变更对项目影响的技术。第18章介绍需求跟踪能力怎样把与单个需求同它们的来源和可能产生的未来开发联系起来。第三部分的内容包括一些商业工具的说明，这些工具能增强你管理项目需求的能力。

从原理到实践

要克服障碍，更改旧的传统做法，把新的知识付诸实践的确不是一件容易的事情。你也许仍想保持原有熟悉（可能并不很有效）的方法。为了帮助你改进需求分析，本书的每一章都包括一个称作“下一步”的内容，它细致地教你如何将本章所学的知识真正开始应用于实践。我提供了许多带有详细注释的模板，包括：需求分析文档、审查校对清单列表、需求优先级电子表格等，所有这些都放在我的网站<http://www.processimpact.com>上，希望能帮助你尽快把这些技术应用于实践。请从今天就开始，一点一点地逐渐改进你的需求分析。

一些项目参与者在尝试新需求技术时是很勉强的。有一些人完全是不切实际的，而与这些人合作，所有新技术都是没用的。因此将本书介绍的知识告诉你的同事、客户和管理者，用你们以前项目中遇到的与需求有关的问题或困难来提醒他们，与他们一起讨论这些新技术拥有的潜在优势，共同学习、共同提高。

你不一定非要在一个新项目中开始应用这种改进的需求工程方法。其实就地改变控制过程就是很好的开端。也就是说，你可以从管理需求变更的请求开始，采用这种比过去更好的方法。因为你能开始做系统的影响分析，创建跟踪能力矩阵，从而把新的需求与相应的设计、代码、测试用例都联系起来。为现存系统回头重新编写整个系统的需求规格说明是不现实

的。但你可以为需求写一个更加结构化的新版本，作一些新特点的分析模型，并调查新的需求。逐渐实施改进的需求，其风险较低，并且也可以为你将新技术应用于下一个主要项目奠定基础。

需求工程的目标是做出高质量而并非完美的需求分析，这使得你能在一个可接受的风险限度上实施需求分析。为了把返工重做、不被接收的产品及超期未完成这些风险降低到最小程度，你必须花大量的时间来研究需求工程。而本书将有助于确定何时能达到合适的需求分析质量标准，同时还介绍了具体实施方法。

目 录

译者序
前言

第一部分 软件需求：是什么和为什么

第1章 基本的软件需求	1
1.1 软件需求的定义	2
1.1.1 一些关于“需求”的解释	2
1.1.2 需求的层次	3
1.2 每个项目都有需求	4
1.3 什么情况将会导致好的群体发生不合格的需求说明	5
1.4 高质量的需求过程带来的好处	7
1.5 优秀需求具有的特性	7
1.5.1 需求说明的特征	7
1.5.2 需求规格说明的特点	8
1.6 需求的开发和管理	9
第2章 客户的需求观	11
2.1 谁是客户	12
2.2 客户与开发人员之间的合作关系	12
2.2.1 软件客户需求权利书	13
2.2.2 软件客户需求义务书	15
2.3 “签约”意味着什么	17
第3章 需求工程的推荐方法	18
3.1 知识技能	19
3.2 需求获取	20
3.3 需求分析	21
3.4 需求规格说明	22
3.5 需求验证	23
3.6 需求管理	23
3.7 项目管理	24
第4章 改进需求过程	26
4.1 需求与其他项目过程的联系	26
4.2 软件需求对其他项目风险承担者的影响	27
4.3 软件过程改进的基础	28
4.4 过程改进周期	29
4.4.1 评估当前采用的方法	29

4.4.2 制定改进活动计划	30
4.4.3 建立、实验和实施新的过程	31
4.4.4 评估结果	32
4.5 需求过程的积累材料	33
4.5.1 需求开发过程的积累材料	34
4.5.2 需求管理过程的积累材料	34
4.6 需求过程改进路标	35
第5章 软件需求与风险管理	37
5.1 软件风险管理基础	38
5.1.1 风险管理的要素	38
5.1.2 编写项目风险文档	39
5.1.3 制定风险管理计划	40
5.2 与需求有关的风险	41
5.2.1 需求获取	41
5.2.2 需求分析	42
5.2.3 需求规格说明	42
5.2.4 需求验证	43
5.2.5 需求管理	43
5.3 风险管理是你的好助手	43

第二部分 软件需求工程

第6章 建立项目视图与范围	45
6.1 通过业务需求确定项目视图	45
6.2 项目视图和范围的文档	46
6.3 关联图	50
6.4 把注意力始终集中在项目的范围上	51
第7章 寻找客户的需求	52
7.1 需求的来源	52
7.2 用户类	53
7.3 寻找用户代表	54
7.4 产品的代表者	55
7.4.1 寻求产品代表者	56
7.4.2 产品代表者的期望	56
7.4.3 多个产品代表者	57
7.5 谁作出决策	58
第8章 聆听客户的需求	60

18.1.3 需求跟踪能力工具	153	19.1 使用需求管理工具的益处	159
18.1.4 需求跟踪能力过程	153	19.2 商业需求管理工具	160
18.1.5 需求跟踪能力可行吗, 必要吗? ...	154	19.3 实现需求管理自动化	161
18.2 变更需求代价: 影响分析	154	附录 当前需求实践的自我评估	163
18.2.1 影响分析过程	155	参考文献	167
18.2.2 影响分析报告模板	157	后记	171
第19章 需求管理工具	158		

第一部分 软件需求： 是什么和为什么

第1章 基本的软件需求

“喂，是Phil吗？我是人力资源部的Maria，我们在使用你编写的职员系统时遇到一个问题，一个职员想把她的名字改成Sparkle Starlight，而系统不允许，你能帮帮忙吗？”

“她嫁给了一个姓Starlight的人吗？”Phil问道。

“不，她没有结婚，而仅仅是要更改她的名字，”Maria回答。“就是这问题，好像我们只能在婚姻状况改变时才能更改姓名。”

“当然是这样，我从没想过谁会莫名其妙地更改自己的姓名。我不记得你曾告诉我系统需要处理这样的事情，这就是为什么你们只能在改变婚姻状况对话框中才能进入更改姓名的对话框。”Phil说。

Maria说：“我想你当然知道每个人只要愿意都可以随时合法更改他（她）们的姓名。但不管怎样，我们希望在下周五之前解决这个问题，否则，Sparkle将不能支付她的账单。你能在此前修改好这个错误吗？”

“这并不是我的错！我从来不知道你需要处理这种情况。我现在正忙着做一个新的性能检测系统，并且还要处理职员系统的一些需求变更请求”（传来翻阅稿纸的声音）。“我还有别的事。我只可能在月底前修改好，一周内不行，很抱歉。下次若有类似情况，请早一些告诉我并把它们写下来。”

“那我怎么跟Sparkle说呢？”Maria追问道，“如果她不能支付账单，那她只能挂帐了。”

“Maria，你要明白，这不是我的过错。”Phil坚持道，“如果你一开始就告诉我，你要能随时改变某个人的名字，那这些都不会发生。因此你不能因我未猜出你的想法（需求）就责备我。”

Maria不得不愤怒地屈从：“好吧，好吧，这种烦人的事使我恨死计算机系统了。等你修改好了，马上打电话告诉我，行吧？”

如果作为客户有过类似的经验，你一定知道：一个不能进行一项基本操作的软件产品是多么令人烦恼。尽管开发者最终会满足你的要求，你也不会感谢他。但从开发者角度来看，在整个系统已经完成，用户再提出对功能的进一步要求是多么烦人的事。同时，修改系统的请求迫使你放下当前的项目，而且往往修改请求还要求你优先处理，也是令人很不愉快的。

其实，在软件开发中遇到的许多问题，都是由于收集、编写、协商、修改产品需求过程中的程序和作法（方法）失误带来的。例如上面的Phil和Maria，出现的问题涉及到非正式信息的收集，未确定的或不明确的功能，未发现或未经交流的假设，不完善的需求文档，以及突发的需求变更过程。

对大多数人来说，若要建一幢20万美元的房子，他一定会与建房者详细讨论各种细节，

他们都明白完工以后的修改会造成损失，以及变更细节的危害性。然而，涉及到软件开发，人们却变得“大大咧咧”起来。软件项目中百分之四十至百分之六十的问题都是在需求分析阶段埋下的“祸根”（Leffingwell 1997）。可许多组织仍在那些基本的项目功能上采用一些不合规范的方法，这样导致的后果便是一条鸿沟（期望差异）——开发者开发的与用户所想得到的软件存在着巨大期望差异。

在软件工程中，所有的风险承担者（stakeholder）都感兴趣的就是需求分析阶段。这些风险承担者包括客户、用户、业务或需求分析员（负责收集客户需求并编写文档，以及负责客户与开发机构之间联系沟通的人）、开发人员、测试人员、用户文档编写者、项目管理者 and 客户管理者。这部分工作若处理好了，能开发出很出色的产品，同时会使客户感到满意，开发者也倍感满足、充实。若处理不好，则会导致误解、挫折、障碍以及潜在质量和业务价值上的威胁。因为需求分析奠定了软件工程和项目管理的基础，所以所有风险承担者最好是采用本书提供的有效的需求分析过程。

本章将帮助你：

- 了解软件需求开发中使用的一些关键名词。
- 警惕在软件项目中可能出现的与需求相关的一些问题。
- 知道优秀的需求规格说明应该具有的特点。
- 明白需求开发与需求管理之间的区别。

1.1 软件需求的定义

软件产业存在的一个问题就是缺乏统一定义的名词术语来描述我们的工作。客户所定义的“需求”对开发者似乎是一个较高层次的产品概念。而开发人员所说的“需求”对用户来说又像是详细设计了。实际上，软件需求包含着多个层次，不同层次的需求从不同角度与不同程度反映着细节问题。

IEEE软件工程标准词汇表（1997年）中定义需求为：

- （1）用户解决问题或达到目标所需的条件或权能（Capability）。
- （2）系统或系统部件要满足合同、标准、规范或其它正式规定文档所需具有的条件或权能。
- （3）一种反映上面（1）或（2）所描述的条件或权能的文档说明。

1.1.1 一些关于“需求”的解释

IEEE公布的定义包括从用户角度（系统的外部行为），以及从开发者角度（一些内部特性）来阐述需求。

关键的问题是一定要编写需求文档。我曾经目睹过一个项目中途更换了所有的开发者，客户被迫与新的需求分析者坐到一起。分析人员说：“我们想与你谈谈你的需求。”客户的第一反应便是：“我已经将我的要求都告诉你们前任了，现在我要的就是给我编一个系统”。而实际上，需求并未编写成文档，因此新的分析人员不得不从头做起。所以如果只有一堆邮件、贴条、会谈过几次或一些零碎的对话，你就确信你已明白用户的需求，那完全是自欺欺人。

另外一种定义认为需求是“用户所需要的并能触发一个程序或系统开发工作的说明”（Jones 1994）。需求分析专家Alan Davis（1993）拓展了这个概念：“从系统外部能发现系统

所具有的满足于用户的特点、功能及属性等”。这些定义强调的是产品是什么样的，而并非产品是怎样设计、构造的。而下面的定义则从用户需要进一步转移到了系统特性（Sommerville and Sawyer 1997）：

需求是……指明必须实现什么的规格说明。它描述了系统的行为、特性或属性，是在开发过程中对系统的约束。

从上面这些不同形式的定义不难发现：并没有一个清晰、毫无二义性的“需求”术语存在，真正的“需求”实际上在人们的脑海中。任何文档形式的需求（例如：需求规格说明）仅是一个模型，一种叙述（Lawrence 1998）。我们需要确保所有项目风险承担者在描述需求的那些名词的理解上务必达成共识。

1.1.2 需求的层次

下面这些定义是需求工程领域中常见术语的定义说明。

软件需求包括三个不同的层次——业务需求、用户需求和功能需求——也包括非功能需求。业务需求（business requirement）反映了组织机构或客户对系统、产品高层次的目标要求，它们在项目视图与范围文档中予以说明。用户需求（user requirement）文档描述了用户使用产品必须要完成的任务，这在使用实例（use case）文档或方案脚本（scenario）说明中予以说明。功能需求（functional requirement）定义了开发人员必须实现的软件功能，使得用户能完成他们的任务，从而满足了业务需求。所谓特性（feature）是指逻辑上相关的功能需求的集合，给用户处理提供能力并满足业务需求。软件需求各组成部分之间的关系如图1-1所示。

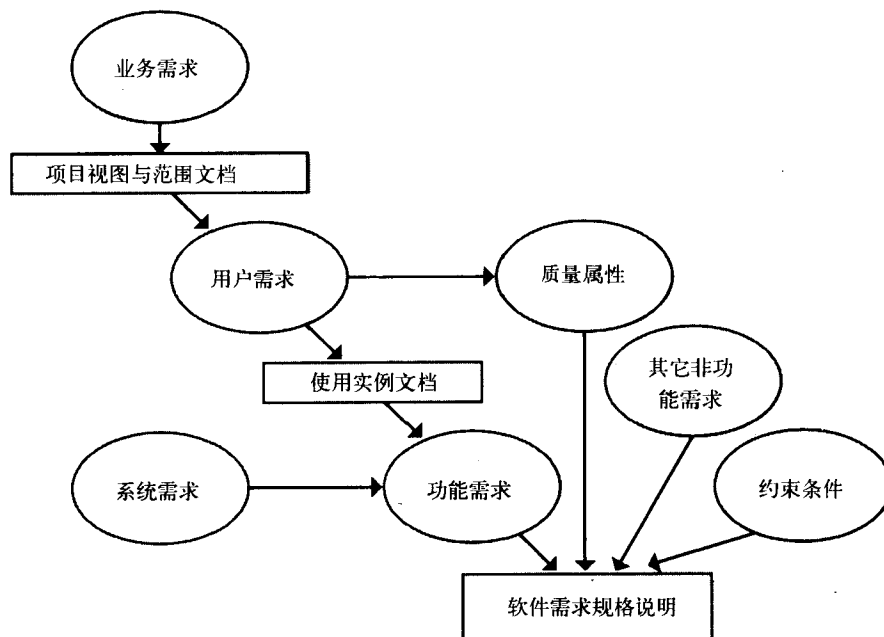


图1-1 软件需求各组成部分之间的关系

在软件需求规格说明（software requirements specification, SRS）中说明的功能需求充分描述了软件系统所应具有的外部行为。软件需求规格说明在开发、测试、质量保证、项目管

理以及相关项目功能中都起了重要的作用。对一个复杂产品来说，软件功能需求也许只是系统需求的一个子集，因为另外一些可能属于软件部件。

作为功能需求的补充，软件需求规格说明还应包括非功能需求，它描述了系统展现给用户的行为和执行的操作等。它包括产品必须遵从的标准、规范和合约；外部界面的具体细节；性能要求；设计或实现的约束条件及质量属性。所谓约束是指对开发人员在软件产品设计和构造上的限制。质量属性是通过多种角度对产品的特点进行描述，从而反映产品功能。多角度描述产品对用户和开发人员都极为重要。

下面以一个字处理程序为例来说明需求的不同种类。业务需求可能是：“用户能有效地纠正文档中的拼写错误”，该产品的包装盒封面上可能会标明这是个满足业务需求的拼写检查器。而对应的用户需求可能是“找出文档中的拼写错误并通过一个提供的替换项列表来供选择替换拼错的词”。同时，该拼写检查器还有许多功能需求，如找到并高亮度提示错词的操作；显示提供替换词的对话框以及实现整个文档范围的替换。

管理人员或市场分析人员会确定软件的业务需求，这使公司运作更加高效（对信息系统而言）或具有很强的市场竞争力（对商业软件产品而言）。所有的用户需求必须与业务需求一致。用户需求使需求分析者能从中总结出功能需求以满足用户对产品的要求从而完成其任务，而开发人员则根据功能需求来设计软件以实现必须的功能。

从以上定义可以发现，需求并未包括设计细节、实现细节、项目计划信息或测试信息。需求与这些没有关系，它关注的是充分说明你究竟想开发什么。项目也有其它方面的需求，如开发环境需求或发布产品及移植到支撑环境的需求。尽管这些需求对项目成功也至关重要，但它们并非本书所要讨论的。

1.2 每个项目都有需求

Frederick Brooks在他1987年的经典的文章“*No Silver Bullet: Essence and Accidents of Software Engineering*”中充分说明了需求过程在软件项目中扮演的重要角色：

开发软件系统最为困难的部分就是准确说明开发什么。最为困难的概念性工作便是编写出详细技术需求，这包括所有面向用户、面向机器和其它软件系统的接口。同时这也是一旦做错，将最终会给系统带来极大损害的部分，并且以后再对它进行修改也极为困难。

每个软件产品都是为了使其用户能以某种方式改善他们的生活，于是，花在了解他们需要上的时间便是使项目成功的一种高层次的投资。这对于商业最终用户应用程序，企业信息系统和软件作为一个大系统的一部分的产品是显而易见的。但是对于我们开发人员来说，并没有编写出客户认可的需求文档，我们如何知道项目于何时结束？而如果我们不知道什么对客户来说是重要的，那我们又如何能使客户感到满意呢？

然而，即便并非出于商业目的的软件需求也是必须的。例如软件库、组件和工具这些供开发小组内部使用的软件。当然你可能偶尔勿需文档说明就能与其他人意见较为一致，但更常见的是出现重复返工这种不可避免的后果，而重新编制代码的代价远远超过重写一份需求文档的代价。

近来，我遇到一个开发小组开发包括流程图工具和源代码编辑器在内的一套计算机辅助软件工程工具。不幸的是，在他们开发完这个工具后，发现这个工具不能打印出源代码文件，而

用户当然希望有这个功能。结果这个小组只好手工抄写源代码文档以供代码检查。这说明那怕需求明确无误并构思准确，如果我们没有编写文档，软件达不到期望目标也只能是咎由自取了。

相反的情况，我曾为一个要集成到“商业错误跟踪系统”中的简单电子邮件界面写了一页需求说明。而Unix系统管理员在为处理电子邮件写脚本时发现简单的一张需求清单竟是如此有用。我依据需求对系统进行测试时，此系统不仅非常清晰地实现了所有必需功能，而且未发现任何错误。

1.3 什么情况将会导致好的群体发生不合格的需求说明

不重视需求过程的项目队伍将自食其果。需求工程中的缺陷将给项目成功带来极大风险，这里的“成功”是指推出的产品能以合理的价格、及时限在功能、质量上完全满足用户的期望。下面将讨论一些需求风险。第5章将介绍怎样应用软件风险管理防止与需求有关的风险的出现。

不适当的需求过程所引起的一些风险

- 用户不多导致产品无法被接受。
- 用户需求增加带来过度的耗费和降低产品的质量。
- 模棱两可的需求说明可能导致时间的浪费和返工。
- 用户增加一些不必要的特性和开发人员画蛇添足(gold-plating)。
- 过分简略的需求说明以致遗漏某些关键需求。
- 忽略某类用户的需求将导致众多客户的不满。
- 不完善的需求说明使得项目计划和跟踪无法准确进行。

1. 无足够用户参与

客户经常不明白为什么收集需求和确保需求质量需花费那么多功夫，开发人员可能也不重视用户的参与。究其原因：一是因为与用户合作不如编写代码有意思；二是因为开发人员觉得已经明白用户的需求了。在某些情况下，与实际使用产品的用户直接接触很困难，而客户也不太明白自己的真正需求。但还是应让具有代表性的用户在项目早期直接参与到开发队伍中，并一同经历整个开发过程。

2. 用户需求的不断增加

在开发中若不断地补充需求，项目就越变越庞大以致超过其计划及预算范围。计划并不总是与项目需求规模与复杂性、风险、开发生产率及需求变更实际情况相一致，这使得问题更难解决。实际上，问题根源在于用户需求的改变和开发者对新需求所作的修改。

要想把需求变更范围控制到最小，必须一开始就对项目视图、范围、目标、约束限制和成功标准给予明确说明，并将此说明作为评价需求变更和新特性的参照框架。说明中包括了对每种变更进行变更影响因素分析的变更控制过程，有助于所有风险承担者明白业务决策的合理性，即为何进行某些变更，相应消耗的时间、资源或特性上的折中。

产品开发中不断延续的变更会使其整体结构日渐紊乱，补丁代码也使得整个程序难以理解和维护。插入补丁代码使模块违背强内聚、松耦合的设计原则，特别是如果项目配置管理工作不完善的话，收回变更和删除特性会带来问题。如果你尽早地区别这些可能带来变更的特性，你就能开发一个更为健壮的结构，并能更好地适应它。这样设计阶段需求变更不会直接导致补丁代码，同时也有利于减少因变更导致质量的下降。

3. 模棱两可的需求

模棱两可是需求规格说明中最为可怕的问题(Lawrence 1996)。它的一层含义是指诸多读者对需求说明产生了不同的理解；另一层含义是指单个读者能用不止一个方式来解释某个需求说明。

模棱两可的需求会使不同的风险承担者产生不同的期望，它会使开发人员为错误问题而浪费时间，并且使测试者与开发者所期望的不一致。一位系统测试人员曾告诉我，她所在的测试组经常对需求理解有误，以致不得不重写许多测试用例并重做许多测试。

模棱两可的需求带来不可避免的后果便是返工——重做一些你认为已做好的事情。返工会耗费开发总费用的40%，而70%~85%的重做是由于需求方面的错误所导致的(Leffingwell 1997)。想像一下如果你能减少一半的返工会是怎样的情况？你能更快地开发出产品，在同样的时间内开发更多、更好的产品，甚至能偶尔回家休息休息。

处理模棱两可需求的一种方法是组织好负责从不同角度审查需求的队伍。仅仅简单浏览一下需求文档是不能解决模棱两可问题的。如果不同的评审者从不同的角度对需求说明给予解释，但每个评审人员都真正了解需求文档，这样二义性就不会直到项目后期才发现，那时再发现的话会使得更正代价很大。其它检测模棱两可需求的技术由Gause和Weinberg(1989)给予介绍，本章的后面也有所涉及。

4. 不必要的特性

“画蛇添足”是指开发人员力图增加一些“用户欣赏”但需求规格说明中并未涉及的新功能。经常发生的情况是用户并不认为这些功能性很有用，以致在其上耗费的努力“白搭”了。开发人员应当为客户构思方案并为他们提供一些具有创新意识的思路，具体提供哪些功能要在客户所需与开发人员在允许时限内的技术可行性之间求得平衡，开发人员应努力使功能简单易用，而不要未经客户同意，擅自脱离客户要求，自作主张。

同样，客户有时也可能要求一些看上去很“酷”，但缺乏实用价值的功能，而实现这些功能只能徒耗时间和成本。为了将“画蛇添足”的危害尽量减小，应确信：你明白为什么要包括这些功能，以及这些功能的“来龙去脉”，这样使得需求分析过程始终是注重那些能使用户完成他们业务任务的核心功能。

5. 过于精简的规格说明

有时，客户并不明白需求分析有如此重要，于是只作一份简略之至的规格说明，仅涉及了产品概念上的内容，然后让开发人员在项目进展中去完善，结果很可能出现的是开发人员先建立产品的结构之后再完成需求说明。这种方法可能适合于尖端研究性的产品或需求本身就十分灵活的情况(McConnell 1996)。但在大多数情况下，这会给开发人员带来挫折(使他们在不正确的假设前提和极其有限的指导下工作)，也会给客户带来烦恼(他们无法得到他们所设想的产品)。

6. 忽略了用户分类

大多数产品是由不同的人使用其不同的特性，使用频繁程度也有所差异，使用者受教育程度和经验水平也不尽相同。如果你不能在项目早期就针对所有这些主要用户进行分类的话，必然导致有的用户对产品感到失望。例如，菜单驱动操作对高级用户太低效了，但含义不清的命令和快捷键又会使不熟练的用户感到困难。

7. 不准确的计划

“上述是我对新产品的看法，好，现在你能告诉我你什么时候能完成吗？”许多开发人员都遇到这种难题。对需求分析缺乏理解会导致过分乐观的估计，而当不可避免的超支发生时，会带来颇多麻烦。据报道，导致需求过程中软件成本估计极不准确的原因主要有以下五点：频繁的需求变更、遗漏的需求、与用户交流不够、质量低下的需求规格说明和不完善的需求分析（Davis 1995）。

对不准确的要求所提问题的正确响应是“等我真正明白你的需求时，我就会来告诉你”。基于不充分信息和未经深思的对需求不成熟的估计很容易为一些因素左右。要作出估计时，最好还是给出一个范围（如最好的情况下，很可能的，最坏情况下）或一个可信赖的程度（我有90%的把握，我能在8周内完成）。未经准备的估计通常是作为一种猜测给出的，听者却认为是一种承诺。因此我们要尽力给出可达到的目标并坚持完成它。

1.4 高质量的需求过程带来的好处

实行有效的需求工程管理的组织能获得多方面的好处。最大的好处是在开发后期和整个维护阶段的重做的工作大大减少了。Boehm（1981）发现要改正在产品付诸应用后所发现的一个需求方面的缺陷比在需求阶段改正这个错误要多付出68倍的成本。近来很多研究表明这种错误导致成本放大因子可以高达200倍。强调需求质量并不能引起某些人的重视，他们错误地认为在需求上消耗多少时间就会导致产品开发推迟多少时间。传统的质量成本角度分析揭示了需求及其它早期质量工作的重要性（Wieggers 1996a）。

正确的需求过程强调产品开发中的通力合作，包括在整个项目过程中多方风险承担者的积极努力。收集需求能使开发小组更好地了解市场，而市场因素是任何项目成功的一个关键因素。在产品开发前了解这些比在遭到客户批评后才意识到要节约很多成本。让用户积极参与需求收集过程能使产品更富有吸引力，而且能拥有忠实的客户关系。通过了解用户的任务需求而不仅仅局限于一些“华丽”的特性，你能避免在无用功能上白耗精力，并且用户的参与能弥补用户期望和开发者实际开发之间的“鸿沟（期望差异）”。

将选定系统的需求明确地分配到各软件子系统，强调采用产品工程的系统方法。这样能简化硬软件的集成，也能确保软硬件系统功能匹配适当。有效的变更控制和影响分析过程也能降低需求变更带来的负面影响。最后，将需求编写成清晰、无二义性的文档将会极大地有利于系统测试，确保产品质量，以使所有风险承担者感到满意。

1.5 优秀需求具有的特性

怎样才能把好的需求规格说明和有问题的需求规格说明区别开来？下面讨论单个需求陈述说明的几个特点（Davis 1993；IEEE 1998）。让风险承担者从不同角度对SRS需求说明进行认真评审，能很好地确定哪些需求确实是需要的。只要你在编写、评审需求时把这些特点记在心中，就会写出更好的（尽管并不十分完美）需求文档，同时也会开发出更好的产品。在第9章中，我们将使用这些特点找到一些需求陈述中的问题并改进之。

1.5.1 需求说明的特征

1. 完整性

每一项需求都必须将所要实现的功能描述清楚，以使开发人员获得设计和实现这些功能