

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名： 杨振祥 日期： 2005.3.30

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在__年解密后适用本授权书。非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名： 杨振祥 日期： 2005.3.30
导师签名： 李茂林 日期： 2005.3.30

基于 DOORS 的软件需求管理过程及改进

摘 要

从上世纪六十年代末出现“软件危机”以来,人们一直在探索提高软件质量,克服危机的方法,从而逐步形成了计算机科学技术领域中的一门新兴学科——计算机软件工程学,通常简称软件工程。

在长期的软件工程研究和实践中,人们逐渐认识到需求对于项目的成功与否起着至关重要的作用。可以说,没有准确的需求就不会有能工作的软件。人们发现,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。于是,对于需求的研究逐渐成为热点课题,形成了软件工程新的子领域——需求工程。人们把整个需求范围称为需求工程。需求工程分为需求开发和需求管理两个部分。需求管理是需求工程的一个重要组成部分。它由4个主要活动组成:变更控制、版本控制、需求跟踪、需求状态跟踪。

通过对需求工程和需求管理的研究,人们提出了许多需求工程方法和工具,一些方法和工具已经比较成熟。尽管如此,现有的需求管理方法和工具离实际应用还有较大的差距。

人们使用需求管理工具来辅助需求管理工作,利用需求作为设计、测试和项目管理的基础。需求管理工具采用数据库来保存需求文本及与需求相关的信息。工具的应用极大地提高了人们的工作效率,并且使需求管理过程更加规范化和精确化。

本文以需求管理过程及其改进为研究重点,以笔者在某大型外企参与需求管理工作的实践为基础,对需求管理过程进行了一定的理论和实践探讨。首先,本文分析总结了需求本身固有的特性——层次性、难于描述性和易变性,从而提出了需求细化和需求级别的概念;其次,本文介绍了在业界应用广泛的需求管理工具——Telelogic DOORS;第三,本文讨论了建立需求数据库的基本原则——层次结构统一化,并提出了一个DOORS需求数据库层次结构及元素命名规范;接着,本文分析了

现有的需求变更管理技术和过程，指出传统的单级变更控制过程的缺点和不足，提出新的需求变更控制过程——两阶段需求变更控制过程，并阐述了在 DOORS 下如何实现这个过程；最后，本文提出需求可跟踪性管理概念，设计需求可跟踪性管理报告模板，并在 DOORS 下编写程序自动生成该需求可跟踪性报告。

关键词：

需求管理 需求细化 需求级别 需求变更控制 需求跟踪 DOORS

SOFTWARE REQUIREMENTS MANAGEMENT PROCESS AND IMPROVEMENT BASED ON DOORS

ABSTRACT

People have been devoted themselves to discovery approaches to conquer the software crisis by improving software quality since the crisis arose in the late sixties of last century. As a result, a new discipline in computer science and technology----computer software engineering methodology, software engineering for short, came up.

In the long-term researches and practices in software engineering people gradually realize that requirements play a vital role in project development. They understand that lack of participation of users, non-complete requirement text and frequent change of requirements are the main reason that information technology projects are not able to be completed under the schedule and budget. Therefore, researches on requirements become hot topics and a sub-field of software engineering----*requirement engineering* arises. Requirement engineering covers the whole scope of requirements. Requirement engineering can be divided into two parts----*requirement development* and *requirement management*. Requirement management is composed of four activities----change control, version control, requirement traceability and requirement state track.

People have developed a lot of methods and tools in requirement engineering, some of which are already rather mature. Even though, we have a long way to go to apply the existing methods and tools of requirement management to practice.

People utilize requirement management tools in work, using requirements as a basis for design, test and project management. Requirement management tools adopt database to store requirement-related information. These tools not only improve efficiency of engineers greatly but also make the requirement management process more standardized and accurate.

The research emphasis of this paper is requirement management process improvement as the title indicates. Firstly, this paper summarizes the natural characteristics of requirement-----layered, difficult to describe and vulnerable to change and then introduces the concept of requirement decomposition and requirement level. Secondly, this paper introduces one of the requirement management tools----Telelogic DOORS which is used world-widely. Thirdly, this paper discusses the basic principle of creating a requirement database, that is, the database must have its requirement hierarchy unified. This paper also develops a schema of requirement database hierarchy and element name convention. Then, this paper analyses the existing requirement change control technology and process, point out the shortcoming of single-layer change control process and develops a new process----two-phase requirement change control process. This paper also depicts how to implement this process in DOORS. Finally, this paper introduces the concept of requirement traceability management, designs a requirement traceability report template in DOORS and develops a DXL script to generate the report automatically.

KEY WORDS:

requirement management, requirement decomposition, requirement level, requirement change control, requirement traceability, DOORS

第一章 绪论

1.1 研究及实践背景

随着软件工程技术的不断发展和实践,需求对于一个项目成功与否的重要性越来越被人们所认识。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”。在长期的软件工程实践中,人们逐步认识到,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。从美国于 1995 年开始的一项调查结果就足以看出这一点。在这项调查中,他们对全国范围内的 8000 个软件项目进行跟踪调查。结果表明,有 1/3 的项目没能完成,而在完成的 2/3 的项目中,又有 1/2 的项目没有成功实施。他们仔细分析失败的原因后发现,与需求过程相关的原因占了 45%,而其中缺乏最终用户的参与以及不完整的需求又是两大首要原因,各占 13% 和 12%。

在软件工程中,所有的风险承担者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、项目管理者和客户管理者。这部分工作处理好了,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。如果处理不好,则会导致误解、挫折、障碍以及潜在的质量和业务价值上的威胁。

随着人们对需求重要性认识的不断加深,对于需求的研究逐渐成为热点课题,形成了软件工程新的子领域——需求工程。人们把整个需求范围称为需求工程。需求工程分为需求开发和需求管理两个部分。需求工程是软件工程中最复杂的过程之一,其复杂性来自于客观和主观两个方面。从客观意义上说,需求工程面对的问题几乎是没有范围的。由于应用领域的广泛性,它的实施无疑与各个应用行业的特征密切相关。其客观上的难度还体现在非功能性需求以及它与功能性需求的错综复杂的联系上,当前对非功能性需求分析建模技术的缺乏大大增加了需求工程的复杂性。从主观意义上说,需求工程需要方方面面人员的参与(如领域专家、领域用户、系统投资人、系统分析员、需求分析员等等),各方面人员有不同的着眼点和不同的知识背景,沟通上的困难给需求工程的实施增加了人为的难度。从当前的研究现状来看,需求工程是系统工程和软件工程的一个交叉分支,涉及到软件系统的目标、软件系统提供的服务、软件系统的约束和软件系统运行的环境。它还涉及这

些因素和系统的精确规格说明以及系统进化之间的关系。它也提供现实需要和软件能力之间的桥梁。

需求管理是需求工程的一个重要组成部分。需求管理由 4 个主要活动组成：变更控制、版本控制、需求跟踪、需求状态跟踪。ISO9001 在 4.3 合同评审中隐含了企业在实施软件开发过程中对需求管理的要求。CMM 成熟度模型是在世界范围内的软件企业中经过实践证明有效的过程改进模型。它在第 2 级就详细地提出了对需求管理过程的要求。相对于 ISO9001，CMM 对需求管理的要求更加具体和细化。而二者的结合可以更好地用于指导软件工程过程中的有效的需求管理过程。

人们在长期的需求管理的研究和实践中，研究和开发出了许多需求管理工具来辅助需求管理工作。相对于传统手工制作文档的方式，工具的应用极大地提高了人们的工作效率，并且使需求管理过程更加规范化和精确化。目前，市场上已存在许多商业需求管理工具，主要有：Caliber-RM、Telelogic DOORS、QSSRequireIT、RequisitePro、RTM Workshop、Vital Link 等。

虽然人们在需求工程和需求管理上的研究已经取得了很大的进展，但与实际应用的距离还比较远，并没有根本解决需求开发和管理过程存在的问题。各个软件开发组织以这些理论为指导，根据自身的实际情况，与所使用的工具相结合，制定符合自身需要的需求开发和管理过程，并在实践中不断改进这些过程。一个成熟组织的软件过程应该是一个不断改进的过程。

本文主要围绕需求管理中的变更管理和需求跟踪技术进行论述，以笔者在某大型外企参与需求管理工作为基础，试图对需求管理技术进行一定的理论探讨，进而提出需求管理过程改进的方法和途径，并利用需求管理工具进行初步实现和验证。

1.2 论文内容及成果

本论文以笔者在某大型外企参与需求管理工作的实践为基础，对需求管理的主要方面——需求变更管理过程和需求跟踪技术进行了一定的理论探讨，特别是提出了新的需求变更管理过程和需求可跟踪性管理报告模板。同时，本论文还简单介绍了 Telelogic 公司的需求管理工具 DOORS，包括它的功能、基本概念和组成结构，在此基础上，将理论探讨的成果在该工具上进行初步实现和验证。

具体来说，本论文的工作及成果包括：

- 分析总结了需求本身固有的特性，即层次性，难于描述性和易变性。在此基础上，提出了需求细化和需求级别的概念。

- 讨论建立需求数据库的基本原则——层次结构统一化，以此为基础，提出了一个 DOORS 需求数据库层次结构及元素命名规范。
- 分析了现有的需求变更管理技术和过程，指出传统单级变更控制过程的缺点和不足，提出新的需求变更控制过程——两阶段需求变更控制过程，并阐述了在 DOORS 下如何实现这个过程。
- 提出需求可跟踪性管理概念，设计需求可跟踪性管理报告模板，并在 DOORS 下编写程序自动生成该需求可跟踪性报告。

1.3 论文的组织

本论文总共分为 7 章。

第一章是绪论，概括介绍了课题的研究背景和本论文的主要内容和成果。

第二章简要介绍了需求、需求工程和需求管理的知识，分析总结了需求本身固有的三个特性，引出了需求细化和需求级别的概念。

第三章简要介绍了一种在业界应用广泛的需求管理工具 Telelogic DOORS，包括它的功能、基本概念和组成结构。

第四章阐述了实现需求管理所要做的第一步，即如何创建需求数据库，提出创建需求数据库应当遵循的层次结构统一化的原则，并提出了一套基于 DOORS 的需求数据库层次结构规范。

第五章分析了现有的需求变更管理过程的缺点和不足，提出新的两阶段需求变更控制过程，并阐述了如何在 DOORS 上实现该过程。

第六章阐述了需求跟踪技术和需求可跟踪性管理概念，讨论了需求可跟踪性管理报告模板的设计以及在 DOORS 下如何实现该需求可跟踪性报告自动生成。

第七章是结束语，是对本论文内容的小结。

第二章 需求管理概述

2.1 需求概述

如第一章所述,需求对于一个项目成功与否的重要性是不言而喻的。事实是,在软件开发中遇到的许多问题,都是由于在收集、编写、协商、修改产品需求过程中的手续和做法失误带来的。因为需求工程奠定了软件工程和项目管理的基础,所以所有风险承担者都必须积极参与到这个工程中来。

2.1.1 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”对于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”对于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和深度反映着细节问题。

IEEE 对需求的定义(IEEE 软件工程标准词汇表,1997)是:

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

IEEE 公布的定义包括从用户角度(系统的外部行为)和开发者角度(一些内部特性)来阐述需求概念。关键的一点是一定要编写需求文档。如果只有一堆邮件、纸条,会谈过几次或一些零星的对话,你就确信你已经明白用户的需求,那完全是自欺欺人。

下面是人们对需求定义的另外一些看法:

- 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,并且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。
- 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:并没有一个清晰的、毫无二义性的“需求”术语存在,真正的“需求”实际上存在于人们的脑海中。任何文档形式的需求

(如需求规格说明书) 仅是一个模型, 一种叙述。我们需要确保所有项目承担者在描述需求的那些名词的理解上务必达成共识。

2.1.2 需求的特性

尽管人们已经逐步认识到需求的重要性, 需求工程技术和方法也得到不断发展, 但是为什么还是有许多软件项目因为需求问题而失败呢? 这除了与开发者对需求工程技术的应用有关外, 还与需求本身所具有的特性紧密相关。正是需求本身的这些特性注定了完成需求工程决不是一件容易的事。

需求具有层次性

软件需求是有层次的, 它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)反映了组织机构或用户对系统、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, 这在使用用例(use case)文档或方案情景(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, 为用户提供处理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 2-1 所示。

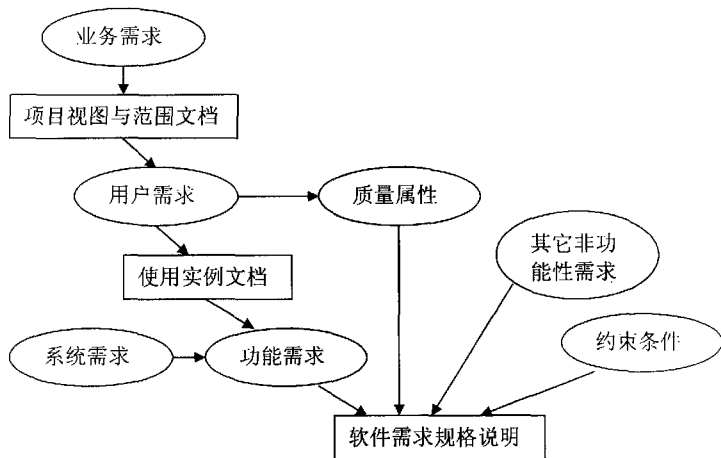


图 2-1 软件需求各组成部分之间的关系

作为补充, 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。值得注意的是, 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。

需求具有难于描述性

需求该如何描述呢? 如果一个系统的需求能够被完整地、毫无二义性地描述清楚, 那么这个系统成功的可能性就大大地提高了。但是, 实际上要做到这一点并不容易。经常出现的情况是, 一个软件开发完毕时却发现它达不到用户的期望, 甚至相差甚远。这可能就是因为用户的期望没有被正确描述的缘故。从上面关于需求定义的讨论中就可以看出, 需求本身具有难于描述的特性, 这个特性将导致项目中担任不同角色之间的沟通面临很大的困难。需求的难于描述表现在以下几个方面:

(1) 问题域本身就是极其复杂的, 尤其对于大型的软件系统来说更是如此。要把这样复杂的系统描述清楚本就不是一件容易的事。

(2) 描述需求的词汇、术语和概念的不一致和过度使用。这些词汇(包括诸如“用例”、“特性”和“需求”这样的基本概念)是“每个人都理解”的日常词汇, 但是在给定的上下文中每个人又赋予了它们自己的含义。这种情形的结果就是沟通不畅。

(3) 在大多数的软件系统中, 最终用户可能都不清楚他的需求是什么。事实上, 这是经常发生的情况。如果你的用户告诉你需求就是这些了, 不要相信他, 继续刨根问底, 直到你们都筋疲力尽了。

需求具有易变性

显然, 需求是不会一成不变的。相反的, 在几乎所有的软件项目开发中, 用户的需求会不断变化, 有时甚至到了让开发人员无法忍受的地步。导致需求变化的因素包括:

(1) 在系统开发期间业务过程、市场机会、竞争产品和软件技术是可能会发生变化的; 需求也必然会发生变化。对许多项目来说, 一些需求的改进是合理的且不可避免。但是即便如此, 也必须对需求变更实施控制。

(2) 需求的难于描述性客观上导致需求的易变。对需求的描述是一个逐步精确的过程, 在这个过程中, 需求必然会发生变化。

(3) 最糟糕的莫过于用户觉得如果不再加些什么功能就对不起自己。这样的心理将导致用户无休止的要求。这些要求有时是一些看上去很“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。

2.1.3 需求的细化

如上一节所述，层次性是需求的固有特性。需求可以分为三个层次，即业务需求，用户需求和功能需求。除此之外，对于大型复杂系统而言，它一般是由若干个子系统构成的，子系统可能再划分为更小的子系统。也就是说，大型系统往往是由不同级别的子系统逐级构成的。因此，软件的功能需求也相应地被细化即分解为不同级别的子系统的功能需求，即每一个功能需求都有一定的需求级别。需求级别的定义和分配要根据具体的项目来定义。一般来说，有多少级别的子系统，就要定义多少个级别的需求。图 2-2 显示了一个 GSM 系统需求级别定义的例子。

级别 0: 市场/用户级	描述客户需求、产品策略、产品特性等
级别 1: GSM 系统级	以系统的观点描述系统级别的需求、模型、接口等
级别 2: 子系统级 (RAN, CORE, OSS)	描述子系统的功能
级别 3: 网元级描述 (BTS, MM, AN, OMCR)	描述网元功能
级别 4: 具体功能级	具体的功能需求描述

图 2-2 一个 GSM 系统需求级别定义的例子

需求细化是需求开发和管理中不可缺少的过程。需求细化就是需求分析逐步求精的过程，是正确定义系统的一个重要方法和手段。在需求细化过程中，相关人员必须保证在不同级别之间的需求进行正确的关联。这些关联是进行需求变更和需求跟踪管理的重要依据。

2.1.4 需求的属性

除了文本，每个功能需求应该有一些与之相关的附加信息或称之为属性。这些属性在它的预期功能性之外为每个需求建立了一个上下文和背景资料。对于大型的复杂项目来说，丰富的属性集合显得尤为重要。

下面列出来的属性是在需求文档中经常需要考虑的：

- 创建需求的时间
- 需求的版本号
- 创建需求的作者
- 负责认可该需求的人员
- 需求的状态
- 需求的原因或根据（或信息的出处）
- 需求涉及的子系统
- 需求涉及的产品版本号
- 使用的验证方法或接受的测试标准
- 需求的优先级
- 需求的级别
- 需求的稳定性

定义和更新这些属性值是需求管理的一部分。精心挑选属性最小子集能帮助你有效管理项目。例如，你不必把负责认可该需求的人员和涉及的子系统都记录下来。如果这样的属性在别的地方已经设置了，在总的开发跟踪系统中就不必在需求数据库中重复设置。

在整个开发过程中，跟踪每个需求的状态是需求管理的另一个重要方面。在每一个可能的状态类别中，如果能周期性地报告各状态类别在整个需求中所占的百分比将会改进项目的监控工作。“需求状态”属性使这项工作变得容易。表 2-1 建议了几种需求状态值。需要注意的是，只有当指定的状态转换条件被严格满足时，才能由专门的负责人员去改变需求状态。

表 2-1 建议的需求状态值表

状 态 值	定 义
已建议	该需求已被有权提出需求的人建议
已批准	该需求已被分析，估计了其对项目余下部分的影响，已用一个确定的产品版本号或创建编号分配到相关的基线中，软件开发团队已同意实现该项需求

表 2-1 建议的需求状态值表（续）

状 态 值	定 义
已实现	已实现需求代码的设计编写和单元测试
已验证	使用所选择的方法已验证了实现的需求，例如测试和检测，审查该需求跟踪与测试用例相符。该需求现在被认为完成
已删除	计划的需求已从基线中删除，但包括一个原因说明和做出闪出决定的人员

2.1.5 如何编写好的需求说明？

现在我们已经知道，好的需求说明对一个项目而言有多么重要。任何产品的质量需要其原始材料的质量保证，糟糕的软件需求说明书不可能产出优秀的软件。那么，什么样的需求描述才是高质量的需求说明呢？

高质量需求描述的特性

(1) 完整性：每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所需的所有必要信息。

(2) 正确性：每个需求必须精确描述要交付的功能。正确性依据于需求的来源，如真实的客户或高级别的系统需求说明书。一个软件需求与其对应的系统需求说明书相抵触是不正确的。只有用户的代表能够决定用户需求的正确性，这就是为什么一定要有用户的积极参与的原因所在。

(3) 可行性：在已知的能力、有限的系统及其环境中每个需求必须是可实现的。为了避免需求的不可行性，在需求分析阶段应该有一个开发人员参与，在抽象阶段应该有市场人员参与。这个开发人员应能检查在技术上什么能做什么不能做，哪些需要需要额外的付出或者和其他的权衡。

(4) 必要性：每一项需求都应当把客户真正需要的和最终系统所需遵从的标准记录下来。“必要性”也可以理解为每项需求都是用来授权你编写文档的“根源”。要使每项需求都能回溯到某项客户的输入，如使用用例或别的来源。

(5) 划分优先级：为了表明在一个详细的产品版本中应包含哪些要点，需要为每个需求，特征，或用例分配实现的优先权。如果所有的需求都被视为同等重要，那么由于在开发中，预算削减，计划超时或组员的离开导致新的需求时，项目经理将不能起到作用。

(6) 无二义性：需求叙述的读者应只能从其得到唯一的解释说明，同样，一个需求的多个读者也应达成共识。自然语言极易导致含糊。要避免使用一些对于 SRS 作者很清楚但对于读者不清楚的主观词汇，如：用户友好性，容易，简单，快速，有

效，几个，艺术级，改善的，最大，最小等等。每写一个需要都应简洁，简单，直观的采用用户熟知的语言，不要采用计算机术语。检查需求模糊的有效方式包括需求说明书的正规检查，根据需求写测试，开发原型以及设计特定的情景模式。

(7) 可证实：看你是否能够做出测试计划或其他验证方式，如检查和实证，来决定在产品中每个需求是否正确的实现。如果需求是不可验证的，决定需求是不是正确的实现就成了判断的事。需求之间不一致，不可行，不明确也能导致不可证实。任何需求如果说产品将要支持什么也是不可证实的。

这些有关需求质量的特性的描述在理论上都是非常好的，但一个好的需求到底应该怎样编写呢？编写优秀的需求是没有公式化的方法的。这需要大量的经验，要从过去的文档中发现的问题学习。下面介绍的是一些经过实践检验的编写需求时必须遵循的原则和方针。

编写高质量需求的原则

(1) 句子和段落要短，要采用主动语气，使用正确的语法，拼写，标点。

(2) 使用术语要保持一致性，并在术语表或数据字典中定义它们。

(3) 正确地把握细化程度，要避免包含多个需求的长的叙述段落，编写独立的可测试的需求。检查该原则是否被遵守的一个方法是：如果一小部分测试可以验证一个需求正确，那么它已经正确的细化了；如果需要多种不同类的测试，则说明可能几个需求被挤到一起了，需要进一步进行细化。

(4) 避免将多个需求合成了单个需求。不要在一个需求中使用“和”/“或”。

(5) 通篇文档细节上要保持一致。

(6) 避免在需求说明书中重复包含相同的需求，这样虽然可以使文档更易于阅读，但也会给文档的维护增加困难。文档的多份文本要在同一时间内全部更新，避免不一致性。

如果你遵从了这些原则，你能够尽早地经常正式或非正式的审查需求，这些需求对于产品的构造，系统测试以及最后的客户满意，都会成为好的基石。并且要记住，没有高质量的需求，软件就象一盒巧克力，你永远不知道你会得到什么。

2.2 需求工程与需求管理

2.2.1 什么是需求工程

需求工程是随着计算机的发展而发展的。在计算机发展的初期，软件规模不大，软件开发所关注的是代码编写，需求分析很少受到重视。后来软件开发引入了

生命周期的概念,需求分析成为其第一阶段。随着软件系统规模的扩大,需求分析与定义在整个软件开发与维护过程中越来越重要,直接关系到软件的成功与否。人们逐渐认识到需求分析活动不再仅限于软件开发的最初阶段,它贯穿于系统开发的整个生命周期。80年代中期,形成了软件工程的子领域——需求工程(requirement engineering, RE)。进入90年代以来,需求工程成为研究的热点之一。从1993年起每两年举办一次需求工程国际研讨会(ISRE),自1994年起每两年举办一次需求工程国际会议(ICRE),1996年Springer-Verlag发行了一新的刊物——《Requirements Engineering》。一些关于需求工程的工作小组也相继成立,如欧洲的RENOIR (Requirements Engineering Network of International Cooperating Research Groups),并开始开展工作。

人们把整个需求范围称为需求工程。需求工程的目的是通过与用户广泛地交流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,促使目标系统最大地满足用户需求。

需求工程是一个不断反复的需求定义、记录和演进的过程,并在最终达到需求的冻结。需求工程划分为需求开发和需求管理两部分,其中需求开发可进一步分为问题获取(elicitation)、分析(analysis)、编写规格说明(specification)和验证(verification)四个阶段,如图2-3所示。

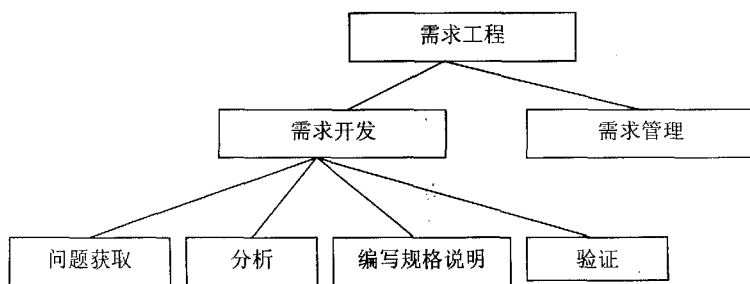


图 2-3 需求工程活动层次分解示意图

需求管理是需求工程生命周期中的重要一环,它负责“建立并维护在软件工程中同客户达成的合同”,力图实现最终产品同需求性的最佳结合。这种合同都包含在编写的需求说明书与模型中。通常的需求管理活动包括:

- 定义需求基线(baseline, 需求文档的主体)。
- 评审提出的需求变更, 评估每项变更的可能影响从而决定是否实施它。
- 以一种可控制的方式将需求变更融入到项目中。
- 使当前的项目计划与需求一致。
- 估计变更需求所产生的影响并在此基础上协商新的承诺(约定)。
- 让每项需求都能与其对应的设计、源代码和测试用例联系起来已实现跟踪。
- 在整个项目过程中跟踪需求状态及其变更情况。

图 2-4 从另一个角度显示了需求开发与需求管理之间的区别。

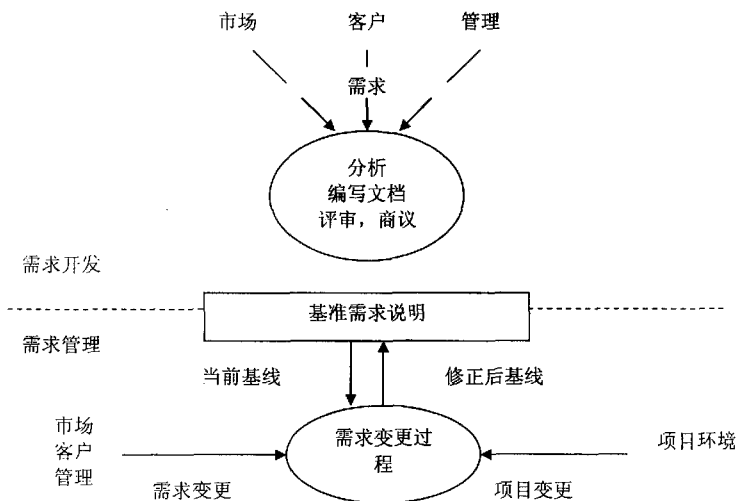


图 2-4 需求开发与需求管理之间的界限

2.2.2 什么是需求管理

如上节所述, 需求开发和需求管理是需求工程的两个不同的组成部分。而连接它们的桥梁就是需求基线(baseline)。这个基线在客户和开发人员之间就构筑了计划产品功能需求和非功能需求的一个约定(agreement)。需求管理包括在工程进展过程中维持需求约定集成性和精确性的所有活动。需求管理强调:

- 控制对需求基线的变动。
- 保持项目计划与需求一致。
- 控制单个需求与需求文档的版本情况
- 管理需求和联系链之间的联系或管理单个需求和其他项目可交付产品之间的依赖关系。
- 跟踪基线中需求的状态。

图 2-5 显示了需求管理的主要活动。

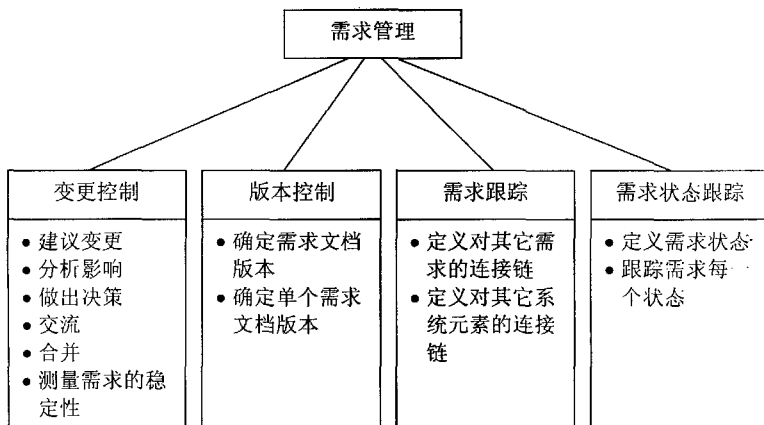


图 2-5 需求管理的主要活动

过程能力成熟度模型(CMM)对需求管理是一个有用的指导。在 CMM2 中，需求管理是其关键过程域 (KPA) 之一，它的目标是：

- (1) 建立软件需求基线供软件工程和管理使用。
- (2) 软件计划、产品和活动同软件需求保持一致。

无论是否知道或关心过程成熟度模型，大多数软件开发组织将会从达成这两个目标中获益。过程成熟度模型确定若干先决条件和技术策略，使组织能持续地达到这两个目标，但并不指定组织必须遵循的需求管理过程。

需求管理的关键过程域不涉及收集和分析项目需求，而是假定已收集了软件需求或已由高一级的系统给定了需求。这与需求开发和需求管理的界限划分是吻合的。

关键过程域同样建议通过版本控制和变更控制来管理需求文档。版本控制确保随时能知道在开发和计划中正在使用的需求的版本情况。变更控制提供了支配下的规范的方式来统一需求变更，并且基于业务和技术的因素来同意或反对建议的变更。

第三章 需求管理工具

3.1 需求管理工具概述

3.1.1 基于文档存储需求的局限性

在计算机软件发展的初期，软件规模很小，那时候还没有出现需求管理工具，人们采用文档文件的方式来存储软件需求说明和其它文档。今天在一些小规模的软件系统开发中，人们也还这样做。但是，随着计算机技术的发展，各种计算机应用系统越来越复杂，软件规模也越来越庞大。这时传统的基于文档文件存储需求的方式越来越显露出它的局限性，主要体现在：

- 手工维护大量文档文件十分困难。
- 很难保持文档与现实的一致。
- 通知受变更影响的设计人员是手工过程。
- 不太容易做到为每一个需求保存附加的信息。
- 很难在功能需求与相应的用例、设计、代码、测试和项目任务之间建立联系链。
- 很难跟踪每个需求的状态。
- 异地协作开发变得很困难。

随着软件工程技术的发展，需求管理的任务越来越繁重，人们迫切需要使用自动化的需求管理工具来提高工作效率。

3.1.2 需求管理工具简介

需求管理工具使用多用户数据库来保存与需求相关的信息，让你不必担心传统方式下存在的问题。小一点的项目可以使用电子表格或简单的数据库管理需求，既保存需求文本，又保存它的几个属性。大项目则可以使用商业需求管理工具，其中包括让用户从源文档中产生需求，定义属性值，操作和显示数据库内容，让需求以各式各样的形式表现出来，定义跟踪能力联系链，让需求同其它软件开发工具相连以协同工作等功能。

需求管理工具不是需求开发工具。这些工具不会帮助你确认未来的客户或者从项目中获得正确的需求。然而，你可以获得许多灵活性，可用来在整个开发期间管理需求的变动，使用需求作为设计、测试、项目管理的基础。这些工具不会代替已

定义的用来描述如何获取和管理需求的处理过程。尽管其他方法同样可以完成工作，但为了高效率就应该使用工具。但记住，不要试图把使用工具作为缺乏方法、训练或理解的补充。

表 3-1 列出了一些这样的商业需求管理工具。本文不打算详细地介绍所有这些工具，也不涉及这些产品之间的横向比较，因为这些工具更新速度较快，而且这不是本文的任务。本文将在下一节详细介绍笔者在工作中使用的工具——Telelogic 公司的 DOORS。这里只做一些简单的介绍。

表 3-1 一些商业的需求管理工具

工 具	所 属 公 司
Caliber-RM	Technology Builders, Inc. (www.tbi.com)
DOORS	Telelogic, Inc. (www.telelogic.com)
QSSRequireIT	Quality Systems and Software, Inc (www.qssrequireit.com)
RequisitePro	Rational Software Corporation (www.rational.com)
RTM Workshop	Integrated Chipware, Inc (www.chipware.com)
Vital Link	Compliance Automation, Inc (www.complianceautomation.com)

需求管理工具允许定义不同种类的数据库元素，例如业务需求、使用实例、功能需求、硬件需求、非功能性需求和测试。这样就可以区分软件需求规格说明中的需求对象及其它有用信息。所有的工具提供了强大的功能来定义每类需求的属性，这一点是它们相对于基于文本的软件需求规格说明方法的强大优势。

绝大多数需求管理工具都在某种程度上与 Word 集成，典型的方式是在 Word 上添加相应的工具条。高级的工具提供丰富的输入、输出文件格式。工具对每个需求不仅有统一的内部标识符，还支持层次编码的数字标签。这些标识符通常是一个短文本字首，例如 UR（代表用户需求，User Requirement），后面再跟一个唯一的整数。高级的工具提供类似于 Windows 资源管理器的层次显示方法用来操作需求树。Telelogic 公司的 DOORS 就是这样的工具。

所有的工具都有在需求同其他系统元素间定义联系链的强大功能。工具的输出能力包括以用户定义的格式或表单报告格式生成需求文档的能力。需求管理工具的其他特点还包括：建立用户小组，定义用户或用户组对项目、需求、属性和属性值的读、写、创建和删除的权限。这些产品还展示了在应用开发中同其他工具（例如：建模，设计，测试，问题跟踪，项目管理工具）相集成的趋势。

3.1.3 使用需求管理工具的益处

在开发过程中，自动化的需求管理工具将极大地帮助开发人员管理需求。特别是随着开发的进行，开发组成员慢慢记不清需求细节，这时商业管理需求工具就变得十分有用。以下是这些工具可以帮你执行的任务：

(1) 管理版本和变更

项目应定义需求基线，基线是每个版本所包括的需求的集合。一些需求管理工具提供灵活的设定基线功能。这些工具可以自动维护每个需求的变动历史，这比手工操作要优越得多。利用这些工具人们可以记录变更决定的基本原则并可根据需要返回到以前的需求版本。通常这些工具包括一个内建的变更建议系统，它可以与变更请求所涉及的需求直接联系。

(2) 存储需求属性

对每一个需求应该保存一些属性，有关人员应能看到这些属性，选择合适的人员更新这些属性值。需求管理工具产生几个系统定义的属性（例如，需求创建日期和创建者），同时允许定义不同类型的其它属性。可以通过排序，过滤，查询数据库来显示满足属性要求的需求子集。

(3) 帮助进行影响分析

通过定义不同种类的需求，子系统的需求，单个子系统和相关系统部件——例如：用例、设计、代码和测试——等各个部分之间的联系链，工具可以确保需求跟踪。联系链可以帮助用来对特定需求所做的变更进行影响分析，即通过确定影响涉及的系统部件来做到这一点。最好的是这些工具可以追溯到功能需求的来源。

(4) 跟踪需求状态

利用数据库保存需求可以很容易知道某个产品包含的所有需求。在开发中跟踪每个需求的状态将可以支持项目的全程跟踪。当项目管理者知道某个项目的下一版本中的百分之五十五的需求已经验证过了，百分之二十八已经实现但还没有验证，百分之十七还没有实现时，他就对项目状况有了很好的了解。

(5) 访问控制

可以对个人、用户小组确定访问权限。绝大多数工具允许共享需求信息，对于地域上分散的组可以通过 Web 网页使用数据库。数据库在需求这一级别通过锁机制进行多用户管理。

(6) 与风险承担者进行沟通

典型的需求管理工具允许小组成员通过多线索电子对话讨论需求。当讨论达成一个新的结果时或某个需求修改后，自动电子邮件系统就会通知涉及的人员。

(7) 重用需求

由于在数据库中保存了需求，在其他项目或子项目中重用需求变为可能。还可以避免信息冗余。

3.2 Telelogic DOORS 概述

DOORS 是全球领先的需求管理工具，目前在全球已有超过 50,000 个用户和 1,000 多家公司在使用。DOORS 可以捕获、连接、跟踪、分析和管理工作需求信息以确保实施的工程与需求规格说明和标准相一致。DOORS 是企业设计的，它具有强大的可伸缩的管理能力，支持多平台操作。

DOORS 可以提供强大的需求管理功能，使你能够沟通商业需求，支持多用户并行工作的方式，提供管理大型复杂项目的能力并验证系统本身的正确性及系统实施的正确性。

DOORS 浏览器视图提供了强大而熟悉的浏览机制，鱼眼（fish-eye）视图可以突出重点地显示，它也可以用色彩表达属性的优先级或试验结果。在市场上没有其它工具可以同时提供这两种图形显示方式。

图 3-1 显示了 DOORS 管理器(DOORS Explorer)界面。

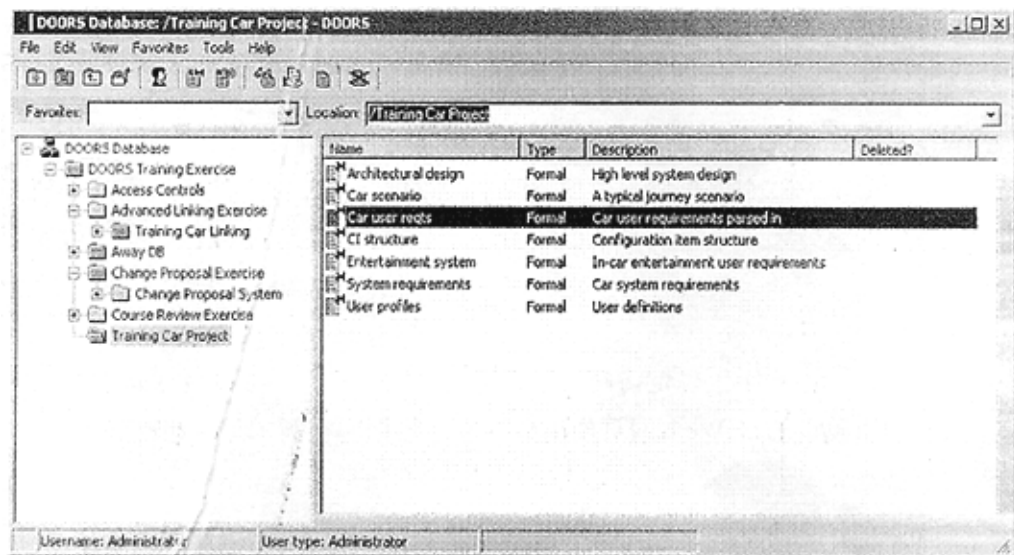
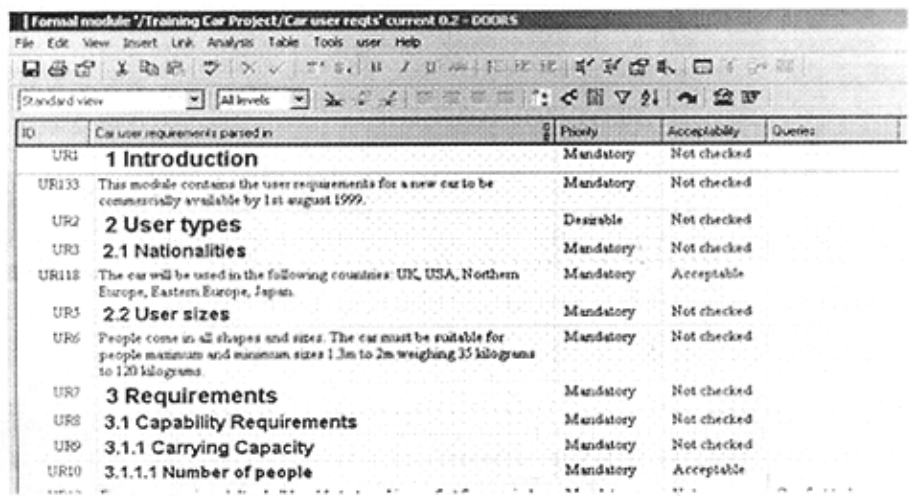


图 3-1 DOORS Explorer 界面

3.2.1 功能及特性

DOORS 具有强大的需求管理功能，这些功能为 DOORS 赢得了其它需求管理工具无可比拟的优势。包括：

(1) DOORS 直观的用户界面可以方便地帮助多用户通过网络并行访问，并且能够维护大量的管理对象（需求和关联信息）和连接。提供 fish-eye 和 MS Windows 资源管理器的图形方式管理视图，每一个用户都可以方便地定制他们想要看到的需求信息——使用图形和颜色方便可靠地标识需求信息。DOORS 是唯一提供电子表格风格的面向文档数据视图的需求管理工具，与 Microsoft Word 和 Excel 有很好的集成。图 3-2 显示了 DOORS 的需求模块编辑界面。



ID	Car user requirements passed in	Priority	Acceptability	Queries
UR1	1 Introduction	Mandatory	Not checked	
UR133	This module contains the user requirements for a new car to be commercially available by 1st august 1999.	Mandatory	Not checked	
UR2	2 User types	Desirable	Not checked	
UR3	2.1 Nationalities	Mandatory	Not checked	
UR118	The car will be used in the following countries: UK, USA, Northern Europe, Eastern Europe, Japan.	Mandatory	Acceptable	
UR5	2.2 User sizes	Mandatory	Not checked	
UR6	People come in all shapes and sizes. The car must be suitable for people maximum and minimum sizes 1.3m to 2m weighing 35 kilograms to 120 kilograms.	Mandatory	Not checked	
UR7	3 Requirements	Mandatory	Not checked	
UR8	3.1 Capability Requirements	Mandatory	Not checked	
UR9	3.1.1 Carrying Capacity	Mandatory	Not checked	
UR10	3.1.1.1 Number of people	Mandatory	Acceptable	

图 3-2 DOORS 需求模块编辑界面

(2) DOORS 内置一套变更建议流程和审核系统，使得用户可以对需求递交变更建议，包括理由。内部的项目连接允许项目共享需求、设计和测试，以及提高需求的跟踪能力。讨论机制支持用户针对一个意见进行合作交流，以加快意见或想法的确立、执行、转换和实现。Distributed Data Management (DDM) 分布数据管理支持远程用户临时访问和使用 DOORS 的所有功能。然后再离线工作，并且远程用户可以将数据更新到主数据库中——这使得那些异地的团队成员和子承包商可以方便地合作开发和沟通。

(3) DOORS 为用户提供了无限制关系的、多级的、用户可自定制的跟踪能力，例如：需求到测试、需求到设计、设计到代码、需求到任务和项目计划到角色。

DOORS 的跟踪向导可以如同需求那样跨多级地生成连接报告，并且在相同的视图中显示——提供 fool-proof 周期确认和验证。

(4) DOORS 提供强大的工具支持实现动态的需求管理。动态需求管理是应对项目开发过程中需求演进的解决方案。通过实施动态需求管理，项目开发人员能够始终控制需求的变更，并且满足成本、项目质量和发布时间等条件。这些工具包括：

- 智能跟踪(Intelligent Traceability™)——确保开发人员在项目进展过程中始终使用正确的基线。
- 预期可疑链接(Proactive Suspect Links)——确保将需求变更通知项目组里的每个成员
- 需求变更建议过程(Requirements Change Proposal Process)——提供基于角色的需求组件视图，使用户可以提交、评审和通过需求变更建议。

3.2.2 组织结构及相关概念

本节讨论的是在 DOORS 中需求是如何被组织的，这将涉及到组成 DOORS 的元素和 DOORS 的基本概念。这些元素包括：文件夹(Folder)、项目(Project)、模块(Module)、对象(Object)、属性(Attribute)、链接(Link)和基线(Baseline)。这些元素分别处于不同的层次，下面将一一加以介绍。

- 对象(Object)和属性(Attribute)

对象是一条完整信息的记录。这些信息可以是需求文本、标题、图形和其他任何信息。对应于需求的属性，DOORS 使用属性来存储需求的附加信息。对象的值是由属性的值决定的。DOORS 的属性使用是非常灵活的。人们可以根据项目过程的需要定义符合要求的属性。用户还可以自定义属性的类型。

- 模块(Module)

模块是对象的集合，对应于一个文档。一个大的项目的需求可能需要分为几个子系统，需要由几个文档来共同说明项目需求，也就是需要分为几个模块来描述。模块根据用途不同分为两种：一种称作正规模块(Formal Module)，用来存放需求信息。另一种称作链接模块(Link Module)，用来存放链接(Link)的信息。有关链接的概念将在后面介绍。DOORS 还可以让用户为模块定义不同的视图(View)，使不同的人可以只看到自己感兴趣的信息，提高工作效率。

- 文件夹(Folder)和项目(Project)

就像计算机文件系统中的文件夹一样，DOORS 的文件夹是高层次的容器，它们是模块的集合。项目是一种特殊的文件夹，它对应于实际中的项目。一般来

说，文件夹用于包含不会发生改变的数据，比如规范、标准、模板等等。而项目则用来存放于项目有关的数据，包括需求、用例、设计和测试等。

由以上介绍可以看出，总的来说，DOORS 数据库是由文件夹、项目和模块组成的，如图 3-3 所示。

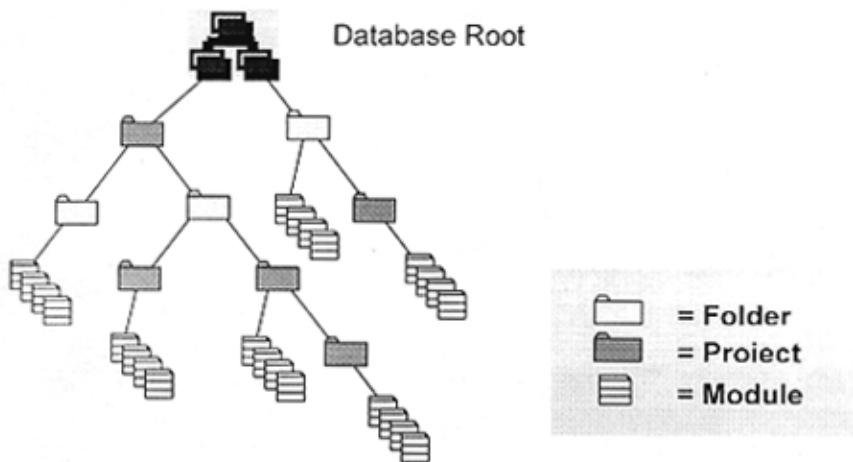


图 3-3 DOORS 的组成结构

下面介绍关于链接(Link)和基线(Baseline)的概念。链接是用来表示和记录 DOORS 中的各种对象（例如：需求、用例、设计、测试）之间的联系链的。基线(Baseline)则是对应于需求管理的基线概念，用来执行需求的版本控制。

- 链接(Link)

对应于需求管理中的联系链，DOORS 使用链接来表示和实现。需求之间的联系链表示需求之间的父层、互连、依赖的关系，根据关系的不同，联系链也有不同的类型。联系链对于实现需求跟踪和变更影响分析是十分重要的必要条件。DOORS 使用链接模块(Link Module)来存储链接，不同类型的链接可以存储在不同的链接模块。对于需求对象来说，链接分为入链接(In-Link)和出链接(Out-Link)两种类型。如图 3-4 所示，一个链接由源模块中的源对象指向目标模块中的目标对象。对源对象来说，这是一个出链接；而对目标对象来说，这是一个入链接。



图 3-4 链接(Link)的概念

● 基线(Baseline)

需求的版本控制是需求管理的一项重要内容。DOORS 使用基线进行版本控制，是这些工作变得简单得多。基线的版本名称又三部分组成：主版本号、次版本号和后缀。

● 视图(View)

与关系数据库中的视图概念一样，DOORS 也提供了视图作为用户管理和操作数据库数据的接口和界面。视图可以包含的信息很多，包括列(Column)信息，过滤器，排序等，甚至窗口的位置，大小信息也可以保存。

3.2.3 脚本开发语言 DXL

DOORS 是一种存储需求的数据库。同其它数据库一样，DOORS 提供自己的脚本语言——DXL(DOORS eXtension Language)供用户使用，允许用户根据自己的需要开发脚本，进一步优化需求管理，提高工作效率。

DXL 的功能十分强大。它被用于实现 DOORS 的许多关键功能，如计算属性值、各种类型文件的导入和导出、影响分析、需求跟踪和联系链管理等。DXL 也可以被用来开发大的插件如 CASE 工具接口和项目管理工具。同时，DXL 还支持强人的图形界面开发，大大加强了 DXL 脚本程序的友好性。使用 DXL 编写的脚本，人们不仅可以项目需要执行常规的或复杂的任务，而且可以设置触发器来响应事件，触发脚本程序执行。人们还可以把开发出来的 DXL 脚本作为子项加入到 DOORS 的菜单中，更加方便操作。

与其它脚本语言一样，DXL 是一种解释性语言。DXL 是一门容易掌握的语言，它在语法上与 C、C++ 很类似，但在使用上更加自由和方便。DXL 的类型十分丰富，除了提供一些基本类型外，如整型、布尔型、字符型和字符串型等，还提供一些内部对象类，如 Folder、Project、Module、Object 和 Attribute 等。这些内置类是用户可以十分方便地操作 DOORS 的文件夹、模块、对象和属性等元素。DXL 不支持自定义类。DXL 还提供了十分丰富的函数库。

第四章 创建需求数据库

4.1 创建需求数据库的原则——层次结构统一化

进行需求管理的第一步是要把需求存储到需求数据库中。这就向我们提出了一个问题，需求在数据库中该如何组织才能使我们更加有效的管理需求，就像我们在 MS Windows Explorer 中如何组织文件才能更好的管理数据一样？

有一点是肯定的，我们必须依据一定的规则在不同的项目之间建立统一的数据层次结构。

4.1.1 统一层次结构的益处

需求数据库具有统一的层次结构有以下好处：

- 有助于规范需求管理过程。创建需求数据库是整个需求管理过程的第一步，因此必须将它纳入到需求管理过程规范中来。这样也使工程师在日常工作中自觉的遵循需求管理过程。
- 为脚本的通用性创造条件。由于规范统一了整个组织的需求数据库结构，因此某一项目开发的脚本工具可以方便地移植应用到其它项目中去，避免重复开发，提高组织工作效率，节约项目成本。
- 提高可操作性，便于管理。由于所有的需求都按照规范约定组织起来，各安其所，操作更加简便，需求管理也就更加容易了。

4.1.2 制定统一层次结构规范的原则

统一层次结构规范的制定必须依据一定的原则，这些原则包括：

- 以组织的需求管理过程为依据。需求数据库的层次结构必须体现组织的需求管理过程，即必须蕴涵着组织的需求管理过程，包括需求细化和需求变更管理过程等。
- 通用性与灵活性相结合。规范必须在组织内具备通用性，这是毫无疑问的，这也是制定规范的出发点之一。同时，规范必须具有灵活性。因为即使是在同一个组织内，项目与项目之间也存在着不同的地方。因此，当将规范应用到项目时，该规范必须是可裁剪的。

- 易于理解 and 实施。这一点是显而易见的。很难想象，一个组织要求工程师去实施一个纷繁复杂的难于理解和实施的规范，这样做只能导致事与愿违罢了。

4.2 DOORS 需求数据库的层次结构规范实例

根据上述制定 DOORS 需求数据库层次结构规范的原则，结合在某一外企公司实习工作的经历，笔者提出一个 DOORS 需求数据库层次结构及元素命名规范，以供读者参考。

本规范实例涉及的内容包括：各种 DOORS 元素（项目、文件夹、模块和属性）等的命名规则及项目层次结构，模块的分类及其结构，下面将一一介绍。

4.2.1 命名规则及项目层次结构

项目(Project)、正规模块(Formal module)和链接模块(Link module)这三种元素的命名必须是唯一的。

(1) 项目命名规则

与文件夹不一样，项目的命名在数据库内是唯一的。在 DOORS 数据库内，项目的命名是由数据库层次结构构成的，就像 MS Windows Explorer 中的绝对路径一样。DOORS 数据库的第一层次被划分为产品开发 (Development) 和沙箱(Sandbox) 两个文件夹。Development 文件夹内包含的是实际产品项目的数据，而 Sandbox 则是用于练习与实验之用。本规范只适用于 Development 文件夹中的项目。数据库的层次结构如下所示：

Development [文件夹元素]

 部门名称 [项目元素]

 项目名称 [项目元素]

 需求级别 [文件夹元素]

 参考信息 [文件夹元素]

 链接 [文件夹元素]

其中，若组织中有多级部门，则部门名称也相应扩展成多级。组织在需求细化过程中将需求细化成几个级别，则就要有几个需求级别文件夹。需求级别文件夹的命名为 LxRS。其中，L 代表 Level，x 是指示级别的数字，RS 是 Requirement Set 的缩写。图 4-1 显示了上述数据库层次结构规则应用的一个例子。

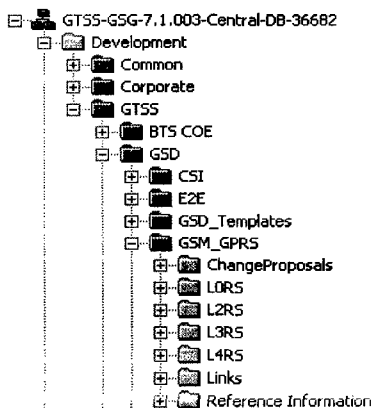


图 4-1 数据库层次规则的应用

(2) 正规模块的分类及命名规则

根据正规模块中存储的数据不同，我们将 Formal Module 分为以下几个类别：

- 市场/用户需求模块——用于存储市场需求或用户需求的模块。
- 技术需求模块——用于存储技术需求包括功能需求的模块。
- 验证模块——用于存储有关需求验证和测试的信息的模块。
- 参考信息模块——用来存储一些与项目相关的参考信息的模块。
- 需求变更请求模块——用于存储需求变更请求的模块。

在一个项目中，同一个类型的正规模块可能不只一个，所以为了保证在同一个项目里正规模块命名的唯一性，正规模块的命名由两个部分组成：

表示类别的前缀+唯一的字符串

其中，“唯一的字符串”是由项目组统一规定的，它应该是能够表示该模块内容含义的字符串。表 4-1 显示了 Formal Module 的命名规则。

表 4-1 Formal Module 的命名规则

正规模块类型	类型前缀	正规模块命名	例子
市场/用户需求模块	MR	MR-唯一命名	<ul style="list-style-type: none"> • MR-Market Rqmts • MR-Customer Rqmts
技术需求模块	TR	TR-唯一命名	<ul style="list-style-type: none"> • TR-Services Manager
验证模块	VER	VER-唯一命名	<ul style="list-style-type: none"> • VER-Test Cases • VER-CDMA I&T

表 4-1 Formal Module 的命名规则 (续)

正规模块类型	类型前缀	正规模块命名	例子
参考信息模块	RI	RI-唯一命名	<ul style="list-style-type: none"> • RI-Standards • RI-Glossary
需求变更请求模块	CR	CR-唯一命名	<ul style="list-style-type: none"> • CR-Sublime MRs

对于处于不同需求级别的技术需求(TR)模块来说,可以在上面的命名规则基础上,再加入指示需求级别的信息。这样,技术需求的命名规则变成:

需求级别+前缀+唯一命名

例如,对于上表中的技术需求模块的例子,假设它的需求级别为 2,那么它的命名更改为: 2TR-Services Manager。

(3) 链接模块的分类及命名规则

对于 Link Module,根据它所存储的链接代表的不同逻辑含义,即所链接的源模块与目的模块不同,我们将 Link Module 分为几种不同类别。与 Formal Module 不同的是,一个项目中的每个类别的 Link Module 只需要一个。因此,不同的链接模块命名也就区分了不同的类别。表 4-2 显示了 Link Module 的分类和每个类别的命名规则。请注意,由于不同的项目有不同的要求,不同的组织其软件过程也不一样,因此这里的链接模块的分类及命名规则仅作为参考。

表 4-2 Link Module 的分类及其命名规则

链接模块命名	链接源模块	链接目的模块
Requirements	TR 模块	MR 模块或上一级 TR 模块
Verification	VER 模块	MR 模块或 TR 模块
Change	CR 模块	MR 模块、TR 模块、RI 模块或 VER 模块
Reference	RI 模块	MR 模块、TR 模块、VER 模块或 CR 模块

4.2.2 正规模块的结构

模块的结构是指模块中的属性定义,包括模块由哪些属性组成,属性的名字和类型,属性的含义和属性用法等内容。

本规范定义的属性可分为两类——必需(Required)的属性和保留(Reserved)的属性。必需的属性是指按照本规范创建的模块必须要具备的属性。保留的属性则不是必需的。但是,如果模块使用了某个保留属性,那么它必须遵循本规范对该属性的定义。

本规范以表格的形式从以下几个方面来说明属性定义：

- 属性名称——属性的名字。
- 属性描述——概括说明属性的含义和使用该属性的好处。
- 属性类型——属性在 DOORS 中所属的类型，如数型、字符串(String)、文本(Text)和枚举(Enumeration)等。如果属性是枚举类型，那么它可以是单值的，也可能是多值的。枚举类型定义的名称将在圆括号中给出。
- 适用的元素——说明该属性是为对象(Object)定义的，或是为整个模块(Module)定义的。
- 使用指导——介绍属性的使用方法指南。为需要缺省值的属性指定其值。如果属性不是由 DOORS 自动产生的，那么这里将标明它是必需的还是保留的属性。

不同类别的正规模块其属性构成结构是不尽相同的，但是有很大的部分是相同的。为了避免不必要的重复，本节先介绍这些公共的属性定义，再依据不同的类别逐一介绍其结构。

(1) 通用的属性定义

下面列出的属性是适用于所有 Formal Module 的。这些属性大多数是系统定义的，即创建模块时由工具自动创建和定义这些属性。

表 4-3 通用的属性定义

属性名称	属性描述	属性类型	适用元素	值与用法
Absolute Number	自动生成，指示对象的绝对编号	Integer	Object	值不重用，即使对象被永久删除
Status	用户自定义的属性，用于指示对象的审核状态	单值的枚举类型(Status List)	Object	必需属性，其值包括：已建议、已批准、已实现、已验证、已删除
Created By	自动生成	String	Module, Object	系统定义的属性，其值是执行操作的用户名
Created On	自动生成	String	Module, Object	系统定义的属性，其值是执行操作的日期
Description	系统定义属性	Text	Module	其值由模块创建者输入
Last Modified By	自动生成	String	Module, Object	其值是最后执行修改的用户名
Last Modified On	自动生成	String	Module, Object	其值是最后执行修改的日期
Legacy No.	使用 DOORS 之前的遗留需求对象编号	Text	Object	保留属性

表 4-3 通用的属性定义 (续)

属性名称	属性描述	属性类型	适用元素	值与用法
Name	系统定义属性, 指示模块名	String	Module	创建模块时输入值
Notes	包含对该需求对象文本的补充支持信息	Text	Object	保留属性
Object Identifier	自动生成, 指示对象标识符	String	Object	其值是模块的前缀属性 (Prefix) 值 + Absolute Number 属性值
Object Level	自动生成, 指示对象的层次	String	Object	系统定义属性
Object Number	自动生成, 指示对象的章节编号	String	Object	系统定义属性
Object Heading	系统定义属性, 指示对象标题	String	Object	一般用来显示章节标题, 应与 Object Text 互斥使用
Object Text	系统定义属性, 对象的主体	Text	Object	对于表示需求的对象它存储需求文本, 应与 Object Heading 互斥使用
Prefix	系统定义属性, 模块前缀	String	Module	创建模块时输入值, 其值必须项目内唯一
Responsible Engineer	指示负责该需求对象的工程师	Text	Object	保留属性

(2) MR 模块的结构

MR 模块包含着市场/客户的需求或者系统的需求。这些需求可以由客户进行验证。下面将介绍 MR 模块特有的属性定义。这些属性只适用于的对象元素。

表 4-4 MR 模块的特有属性定义

属性名称	属性描述	属性类型	值与用法
Allocated To	用于指示该需求细化过程中所分配的下一级别的功能实体或物理实体。	多值的枚举类型 (Allocation List)	保留属性, 只应用于 Object Type 属性值为 Requirement 的对象
Category	用于对需求进行大体的容易理解的分类型。	单值枚举类型 (Category List)	保留属性, 只应用于 Object Type 属性值为 Requirement 的对象
Customer Priority	指示需求的优先级, 只应用于 Object Type 属性值为 Requirement 的对象	单值枚举类型 (Priority List)	保留属性, 取值: High, Medium, Low
Object Type	用于指示对象的功能类型, 使项目人员能够区分需求和其它常用的对象内容的类型	单值枚举类型 (Object Type List)	必需属性, 值的最小集合: Comment, Requirement, Section

表 4-4 MR 模块的特有属性定义 (续)

属性名称	属性描述	属性类型	值与用法
Source	指示该需求创建的原因	单值枚举类型 (Source List)	保留属性
Source Rationale	是对 Source 属性的注解, 说明需求的上下文	Text	保留属性
Technical Feasibility	指示技术可行性分析的结果	单值枚举类型 (Tech Feasibility List)	保留属性

(3) TR 模块的结构

技术需求模块包含实现市场/用户需求的系统的可验证的需求。其中的需求对象用于识别工程需要。下面将介绍 TR 模块特有的属性定义。同样的, 这些属性只适用于的对象元素。

表 4-5 TR 模块的特有属性定义

属性名称	属性描述	属性类型	值与用法
Allocated To	用于指示需求对象在需求细化过程中所对应的下一级别的功能实体或物理实体。	多值的枚举类型 (Allocation List)	必需属性, 只应用于需求对象, 其值由项目决定, 但必须使用“Complete”来指示分配结束
Requirement Level	指示需求对象在需求细化中所处的需求级别	单值枚举类型 (Requirement Level List)	保留属性, 值依项目需求细化过程而定
Feature	用来指示需求对象对应的系统特性	String	保留属性, 可以是特性的简单描述, 也可以是到特性数据库的一个链接
Issue	用于描述与该需求对象相关的问题	Text	保留属性
Object Type	用于指示对象的功能类型, 使项目人员能够区分需求和其它常用的对象内容的类型	单值枚举类型 (Object Type List)	必需属性, 值集合: Comment, Requirement, Section
Parent	用于指示需求对象是否有父亲对象	String	保留属性, 值及含义: Primary—顶级需求对象, 没有父亲 Secondary—从项目约束得到的需求对象, 没有父亲 对于有父亲的对象, 其值为父亲对象 ID

表 4-5 TR 模块的特有属性定义 (续)

属性名称	属性描述	属性类型	值与用法
Release	用于指示包含该需求对象的所有产品版本	多值枚举类型 (Release List)	必需属性, 值依项目而定
Verification Concept	用于描述与该需求对象相关的验证策略	Text	保留属性
Verification Method	指示用于验证该需求对象的方法	多值枚举类型 (Ver Method List)	必需属性
Verification Organization	指示验证该需求对象的人或组织	多值枚举类型 (Ver Org List)	必需属性

(4) VER 模块的结构

VER 模块包含了用于有关需求验证和测试的信息。这些信息包括测试用例、测试脚本, 测试套件, 测试计划等等。

表 4-6 VER 模块的特有属性定义

属性名称	属性描述	属性类型	值与用法
Object Type	指示该验证对象的类型	单值枚举类型 (Object Type List)	保留属性, 值集合: Test Plan, Test Script, Test Case, Model
Test Case ID	用于标识特定的测试用例	String	必需属性
Ver Status	指示该验证对象的状态	单值枚举类型 (Ver Status List)	必需属性。值集合: Passed—用例已执行并通过 Failed—用例已执行未通过 Blocked—用例无法执行 Available—用例未执行 Unavailable—用例还不能执行因为尚有功能未完成
Release	该属性是为模块定义的属性, 用来指示该验证模块对应的产品版本	单值枚举类型 (Release List)	必需属性

(5) RI 模块的结构

RI 模块向其它模块提供附加的信息, 如背景文本, 约束, 标准等等。

表 4-7 RI 模块的特有属性定义

属性名称	属性描述	属性类型	值与用法
Object Type	指示该对象的类型	单值枚举类型 (Object Type List)	保留属性，值依项目而定，例如可以包含 Background, Constraint, Standards 等

(6) CR 模块的结构

表 4-8 CR 模块的特有属性定义

属性名称	属性描述	属性类型	值与用法
Object Type	指示该对象的类型	单值枚举类型 (Object Type List)	必需属性，值集合： Change Request, Section

第五章 需求变更控制过程改进

5.1 需求变更控制概述

5.1.1 需求变更的因素

如果可以一劳永逸地定义系统的一组需求，那么软件的开发就变得容易得多了。可是事实并不那么理想，需求变更从来都是不可避免的。

需求变更有几种原因。有的是内部因素，我们能够控制，但许多是外部因素，是开发人员甚至用户都无法控制的。

(1) 外部因素

外部因素是项目团队难以甚至无法控制的变更的事物。包括：

- 存在项目团队尝试采用新系统解决问题的变更。
- 用户改变了有关他们需要系统完成什么的想法或见解。
- 外部环境发生了变化，产生了新的约束或新的机会。一个最明显的例子是计算机的软硬件条件的不断改善。
- 新系统的产生。新系统的应用导致新需求的产生。

(2) 内部因素

除了外部因素外，还有许多内部因素可以导致变更。

- 在收集需求活动的早期，没有能够在正确的时间向正确的人问正确的问题。
- 没有创建一个实用的过程来帮助管理那些在增量开发中通常会遇到的需求变更，以至于它们不可避免最终爆发。
- 从需求到设计的迭代引发新的需求。

5.1.2 需求变更管理的重要性

正如第二章中所描述的，需求变更管理是需求管理的重要组成部分，是需求管理的一个重要活动之一。易变是需求的固有特性。然而，不被控制的变更使项目陷入混乱，无法按进度和预算完成或者无法保障质量。

做应用软件，很大的一个问题就是前期需求不明确，后期需求变更太多。对需求变更的管理包含在需求管理之中。在软件开发过程中，需求变更不可避免，而且变更有助于最大限度满足客户需求。然而现实中的需求变更管理存在问题。美国

Standish Group 公司对 8400 个软件项目的调查和研究指出三种最经常使项目“遇到困难”的因素，其中，“不断改变的需求和规格说明”占有所有项目的 12%。ESPITI 根据 3800 个调查人的回答，确定在产业中相对重要的软件问题，“管理客户需求”是被调查者回答的两个最大问题之一。由此可见，需求（变更）管理问题是导致软件问题的根本原因之一。

为了使开发组织能够严格控制软件项目，应确保以下事项：

- 仔细评估已建议的变更
- 应挑选合适的人选对变更做出决定
- 变更应及时通知所有涉及的人员
- 项目要按一定的程序来采纳需求变更

5.1.3 变更控制过程

一个好的变更控制过程为项目风险承担者提供了正式的建议需求变更机制。通过这些处理过程，项目负责人可以在信息充分的条件下做出决策，这些决策通过控制产品生存期成本来增加客户和业务价值。开发组织通过变更控制过程来跟踪已建议变更的状态，确保不会丢失或疏忽已建议的变更。一旦确定了一个需求集合的基线，对所有已建议的变更都应该遵循变更控制过程。

变更控制过程并不是给变更设置障碍。相反的，它是一个渠道和过滤器，通过它可以确保采纳最合适的变更，使变更产生的负面影响减少到最小。变更过程应该做成文档，尽可能简单，当然首要的是有效性。如果变更过程没有效率且冗长，又很复杂，大家宁愿用旧方法来做出变更决定。

(1) 角色与责任

表 5-1 列出了在需求变更管理过程中可能存在的不同角色和相应的责任。按项目所处的环境和需要来调整这些角色和相应的责任，在保持有效性的前提下尽可能使过程简单。一个人不必只担任一个角色。

表 5-1 变更管理活动中的项目角色及其责任

角色	角色描述及责任
变更控制委员会主席	变更控制委员会的主席，在 CCB 已经不一致的情况下可以独自做出决定
变更控制委员会 (CCB)	决定采纳或拒绝针对某项目所建议的变更请求的团体
评估者	应项目管理者要求分析所建议的变更带来的影响的人员
修改者	负责实现已被认可的变更请求，按时更新变更状态的人员

表 5-1 变更管理活动中的项目角色及其责任（续）

角色	角色描述及责任
建议者	提交变更请求的人
项目管理者	负责指定评估者和修改者的人员
请求接受者	接受提交变更请求的人
验证者	负责决定变更是否正确执行的人员

(2) 变更请求状态

一个变更请求有一个生存期，相应地有不同的状态。图 5-1 显示了一个变更请求的状态迁移过程。

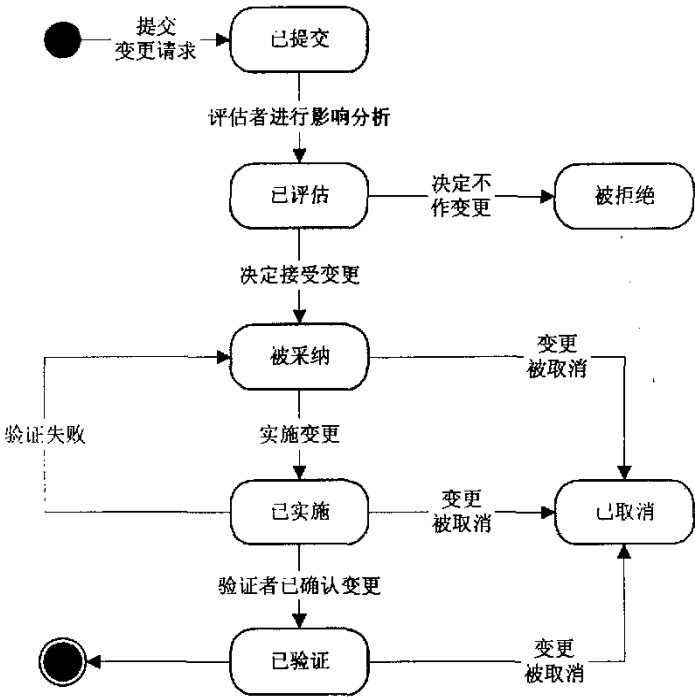


图 5-1 一个变更请求的状态迁移过程

(3) 变更控制委员会

软件开发活动中公认变更控制委员会(CCB)为最好的策略之一。变更控制委员会可以由一个小组担任，也可以由多个不同的组担任。它负责做出决定究竟将哪些已建议的需求变更或新产品特性付诸应用。典型的变更控制委员会同样决定在哪些版本中纠正哪些错误。许多项目已经有负责变更决策的人员，而正式组成变更控制委员会、制定操作步骤会使他们更有效地工作。

广义上, 变更控制委员会对项目任何基线工作产品的变更都可以做出决定, 需求变更只是其中之一。对应于需求的层次性和需求的细化, 可以有不同级别的控制委员会。每个级别的委员会只负责一个需求子集, 例如有些负责业务决策, 另一些负责技术决策。显然, 高级变更控制委员会做出的决策对计划的影响比低级的要大。

以上论述均是从总体上对需求变更管理的介绍, 并未涉及实际的需求变更控制过程。实际上, 不同的组织都根据自身情况和需要制定自己的需求变更控制过程, 许多需求管理工具也提供内置的需求变更控制系统来支持开发组织实现需求变更控制, 例如 DOORS。下面的章节就将介绍 DOORS 提供的需求变更控制过程, 并在它的基础上提出新的变更控制过程。

5.2 两阶段需求变更控制过程

5.2.1 DOORS 内置的变更控制过程——单级需求变更控制过程

如第三章所述, DOORS 内置有一个需求变更控制系统, 称作 Change Proposals System。CPS 系统支持 DOORS 用户对正规模块里的对象进行提交、评审、通过并实施变更建议。它支持的变更控制过程如图 5-2 所示。

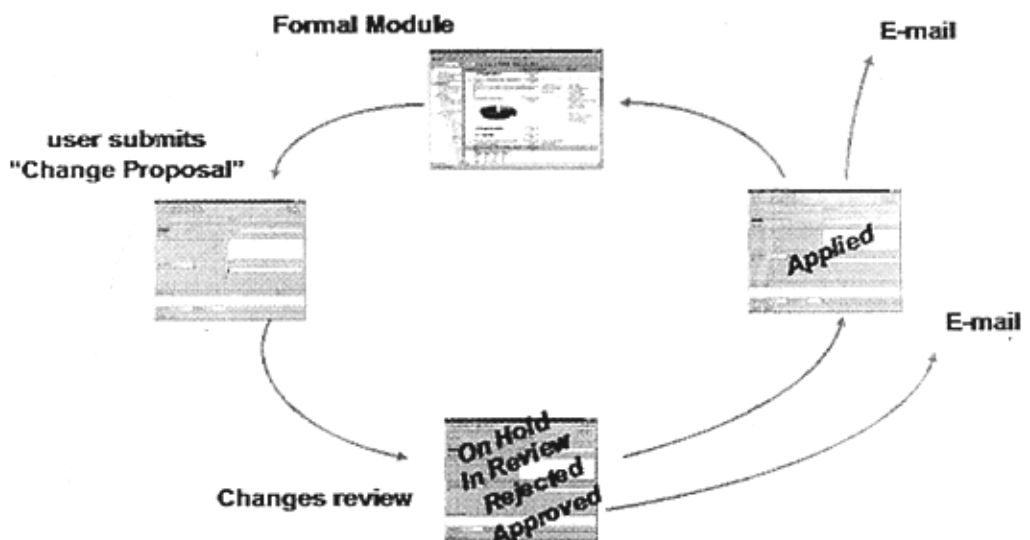


图 5-2 DOORS 需求变更控制过程

DOORS 需求变更管理工作流程:

- (1) 为项目建立 (Setup) CPS 系统
- (2) 提交 (Submit) 变更建议
- (3) 审核 (Review) 变更建议
- (4) 实施 (Apply) 变更建议

DOORS 内置的变更控制过程的特点是只存在单级变更请求, 这些变更请求与需求对象直接相连, 这正是传统需求变更控制过程的做法。本文将这种方式称作单级需求变更控制过程。这种过程具有以下一些缺点和不足:

- 不完整, 不符合实际项目开发工作流程。在实际工作中, 项目开发的不同阶段如设计、编码、测试和客户使用产品阶段都可能涉及到需求的变更。在涉及需求的变更时, 相关人员一开始只能提出变更请求, 而无法确定需要对哪些具体需求进行变更, 这个工作是需要经过需求管理人员经过认真分析才能实现的。
- 变更控制过程不够清晰, 过于简单, 没有区分“是否要改”和“怎样改”两个不同的阶段。

5.2.2 两阶段变更控制过程

针对上述 DOORS 内置的传统变更控制过程的缺点, 本文提出一种新的需求变更控制流程, 称之为两阶段变更控制过程。它把变更控制过程分为两个阶段: 第一阶段是变更请求接受阶段, 解决“要不要变更”的问题; 第二阶段是变更实施阶段, 解决“如何变更”的问题。

如上所述, 导致需求变更的来源存在于项目开发的各个阶段。从某种意义上说, 由客户请求的外部变更是最容易处理的, 因为它们最容易识别, 也最容易找到它们进入项目的途径。但是, 在开发过程中, 还由大量的其它类型的潜在变更。事实上, 许多在设计、编码和测试阶段提出的变更涉及到纠正代码级或设计级的错误, 必须评估这些变更造成的影响, 以确定它们是否与需求有关。许多错误可能影响需求, 要求修改需求或消除某个已知需求的歧义。

因此, 应该采用一种变更请求系统来捕获所有变更请求。该变更请求系统将对所有的变更请求进行影响分析, 生成影响报告, 以确定其是否接受该变更请求, 这就是变更控制过程的第一阶段。经过变更请求系统的分析和过滤, 再把被接受的变更请求送入下一阶段。在该阶段将根据影响分析报告, 对所有受影响的每一项需求

生成变更实施方案。这些方案经过评审后再进行变更实施。这就是变更控制过程的第二阶段。整个变更控制的总体流程如图 5-3 所示。

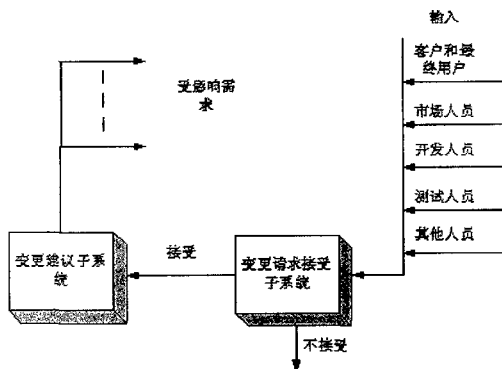


图 5-3 两阶段需求变更控制总体流程

第一阶段 变更请求接受子系统

变更请求接受子系统是需求变更控制过程的第一阶段。在该阶段，相关人员提出变更请求，他们必须对提出变更的原因以及变更请求的优先级进行描述。变更控制委员会将对所有的变更请求进行评估，分析变更可能产生的影响，得到影响分析报告，以确定是否接受该需求变更。

第二阶段 变更建议子系统

变更建议子系统是需求变更控制过程的第二阶段。该阶段将根据第一阶段的影响分析报告，确定受到变更影响的所有层次的需求，制定需求变更实施方案，即需求变更建议，提交变更控制委员会进行评审。这些方案在通过评审后将付诸实施。

图 5-4 显示了更加详细的需求变更控制工作流程。

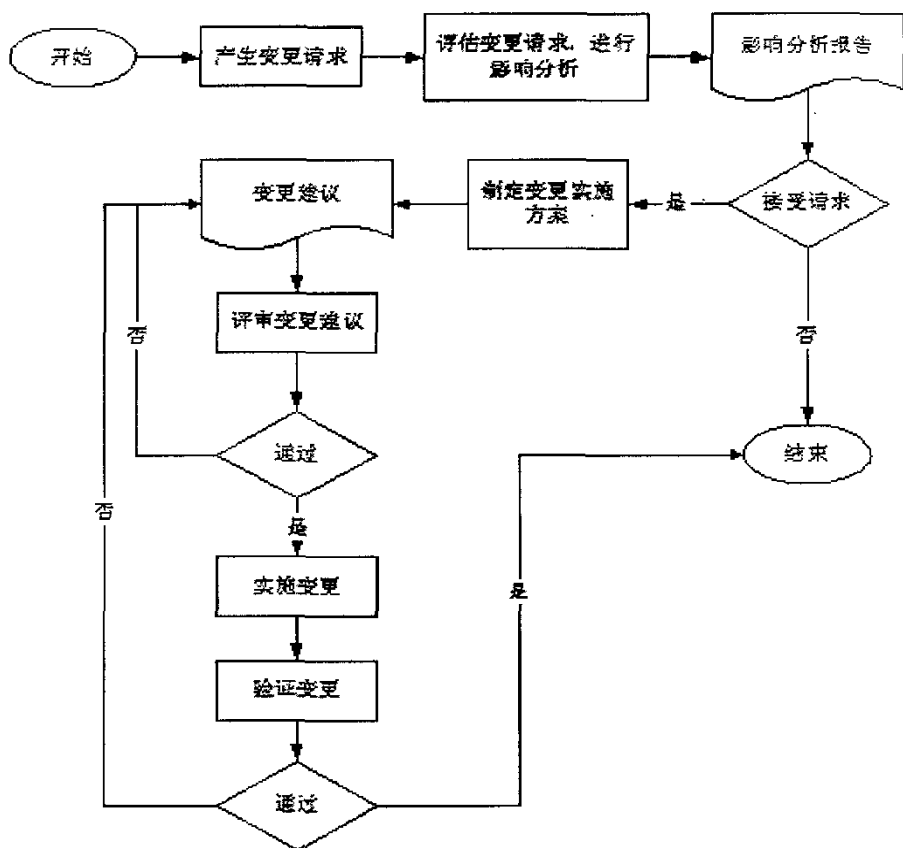


图 5-4 两阶段需求变更控制工作流程

5.2.3 影响分析

许多开发人员有过类似这样的经历：一个表面上很简单的变更转变成很复杂的局面。只要允许需求变更或添加新特性，这种情况就免不了。开发人员往往对建议的软件变更成本或其它衍生结果不一或不能一提供出准确的评估。“变更是免费的”这种误解是造成项目范围延伸的一个原因。人们往往只有在知道变更的成本后才能做出理智的选择。

影响分析是需求变更管理过程的一个重要组成部分。影响分析可以提供对建议的变更的准确理解，帮助做出信息量充分的变更批准决策。通过对变更内容的检验，评估变更对项目范围、项目进度和项目成本的影响，确定对现有的系统做出是修改或抛弃的决定，或者创建新系统以及评估每个任务的工作量。进行影响分析的能力依赖于跟踪能力数据的质量和完整性。

没有人愿意做一个费时费力还要担心意想不到的情况的需求变更。在职业生涯中，绝大多数开发人员会遇到要求添加“没有代价且不影响进度的变更”的要求。对这样令人奇怪的要求的正确回答是“不行，”变更只能在项目时间、预算、资源的限制内进行协商。

5.2.4 两阶段需求变更控制过程的优点

两阶段需求变更控制过程具有如下优点：

- 过程更加清晰，把“是否要改”和“如何改”两个阶段区分开来，更加符合工作实际。
- 将需求变更管理更加细致化，把对各种不同的需求的变更建议通过变更请求联系起来，可以追溯到这些进行这些变更的起源等相关信息，极大地提高需求管理的效率和准确性。
- 人员分工更加明确和合理，工作效率将大大提高。

5.3 基于 DOORS 实现两阶段需求变更控制过程

本节将介绍在 Telelogic DOORS 中如何实现上述两阶段需求变更控制过程。在 DOORS 中，我们创建三种类型的 Formal Module。第一种是存储真正需求的正规模块，我们称这种模块为 Requirements Module。第二种是 Change Requests Module (CR)。它用来存储被需求变更请求，处于需求变更控制过程的第一阶段。第三种是 Change Proposals Module (CP)。该模块用于存储根据影响分析报告生成的对所有受影响需求的变更方案建议。CP 模块并不是唯一的，每一个需求模块都对应一个 CP 模块。每个 CP 模块只存储与之对应的需求模块的变更建议。这三种模块里的对象 (Object)通过链接(Link)联系起来。它们之间的关系如图 5-5 所示。

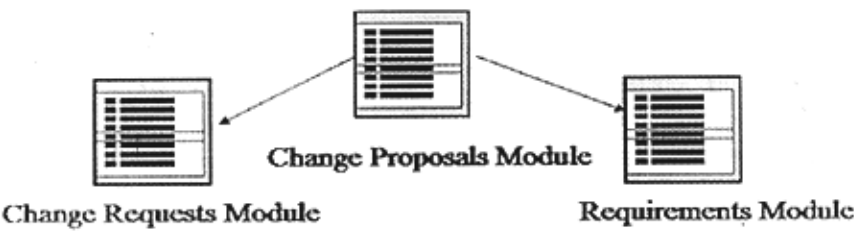


图 5-5 CR、CP 与需求模块之间的关系

CR 模块包含的主要信息 (Attribute 属性) 有：

- 变更请求编号

- 变更请求提交人
- 变更请求提交日期
- 变更原因
- 变更请求的状态
- 变更请求描述

CP 模块包含的主要信息（Attribute 属性）有：

- 变更建议编号
- 变更类型：增加、修改、删除
- 变更原因
- 变更建议描述
- 变更建议提交人
- 变更建议提交日期
- 变更建议的状态
- Proposed XXX 属性：对应需求模块中的 XXX 属性的修改建议
- Using XXX 属性：标志需求模块中的 XXX 属性是否有修改建议

在实际工作中，人们往往将针对某个变更请求(CR)而创建的对某个需求模块的所有变更建议(CP)写在一个 Excel 电子表格文件中，再将文件的内容导入到 DOORS 中去。这种做法的效率将比直接在 DOORS 中进行输入高得多，而且同时生成了一份变更建议的文件拷贝，留待以后必要时进行核对。图 5-6 显示了一个变更建议电子表格的例子。

	A	B	C	D	E	F	G	H	I	J	K
	Change Type	Reason for change	Insertion point	Insertion level	Object Identifier	Object Type	Proposed Object Type	Object Text	Proposed Object Text	Scope	Proposed Scope
1	Modification	Script test modification			U-PM1133	Requirement		The RNC s	The SRNC shall support inverse	SCCP	
2	Added	Script test addition	U-PM865	Same level			Section		Multiplexing over		
3	Added	Script test addition		One level below			Requirement		The SRNC shall support		VS UNI Lnk MeanKbps
4	Added	Script test addition		Same level			Requirement		The SRNC shall support		VS UNI Lnk MeanKbps
5	Added	Script test addition		Same level			Requirement		The SRNC shall support		VS IMA Grp MeanKbps
6	Added	Script test addition		Same level			Requirement		The SRNC shall support		VS IMA Grp MeanKbps
7	Added	Script test addition		Same level			Requirement		The SRNC shall support		VS IMA Lnk MeanKbps
8	Added	Script test addition		Same level			Requirement		The SRNC shall support		VS IMA Lnk MeanKbps
9	Added	Script test addition		Same level			Requirement		The SRNC shall support		VS IMA Lnk MeanKbps

图 5-6 变更建议的电子表格示意图

表格的第一行表示的是各列的标题，比如 Change Type，Reason for change，Insertion point 以及若干 Proposed XXX 列等等。Proposed XXX 列表示的是对需求模

块种的对应属性 XXX 的变更建议, 即将 XXX 属性的值修改成表格中 Proposed XXX 的值。

由 Excel 表格导入到 DOORS 的 CP 模块的工作需要由脚本程序来完成。笔者采用 DOORS 自身提供的脚本语言 DXL 实现了这个程序。图 5-7 显示了该程序的工作流程。

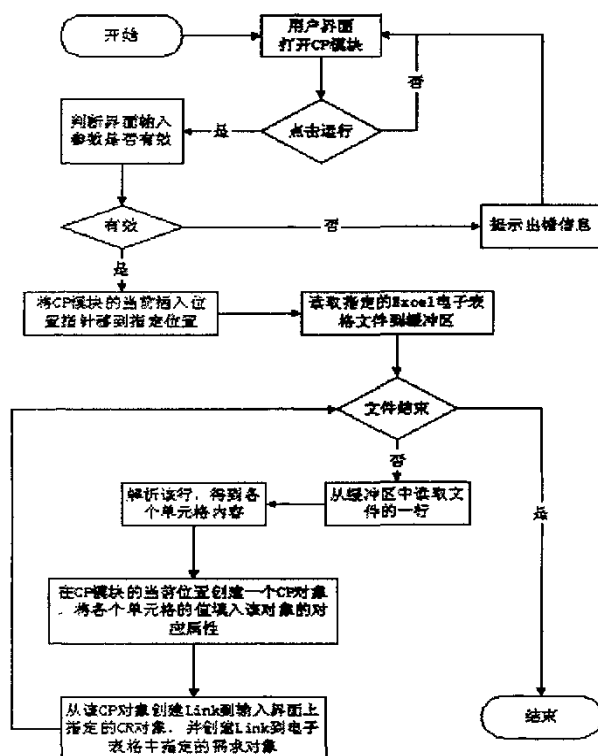


图 5-7 DXL 导入程序的工作流程

该程序的用户界面如图 5-8 所示。

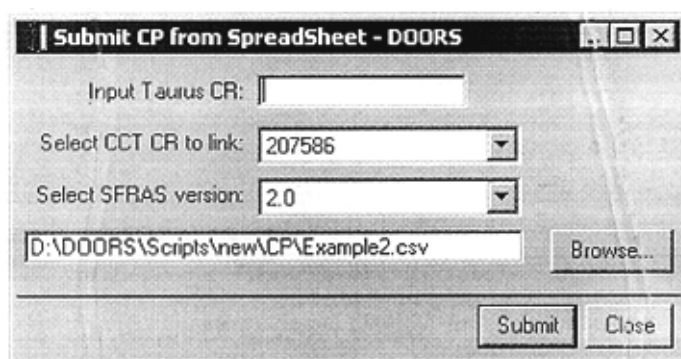


图 5-8 从 Excel 表格导入 CP 的用户界面

第六章 需求跟踪

6.1 概述

需求跟踪包括编制每个需求同系统元素之间的联系文档，这些文档包括别的需求、体系结构，其它设计部件、源代码模块、测试、帮助文件、文档等，跟踪能力使变更影响分析十分便利，有利于确定和评估实现某个建议的需求变更代价。

跟踪能力联系链记录了单个需求之间的父层、互连、依赖的关系。当某个需求变更（被删除或修改）后，这种信息能够确保正确的变更传播，并将相应的任务作出正确的调整。图 6-1 说明了许多能在项目中定义的直接跟踪能力联系链。一个项目不必拥有所有种类的跟踪能力联系链，要根据具体的情况调整。

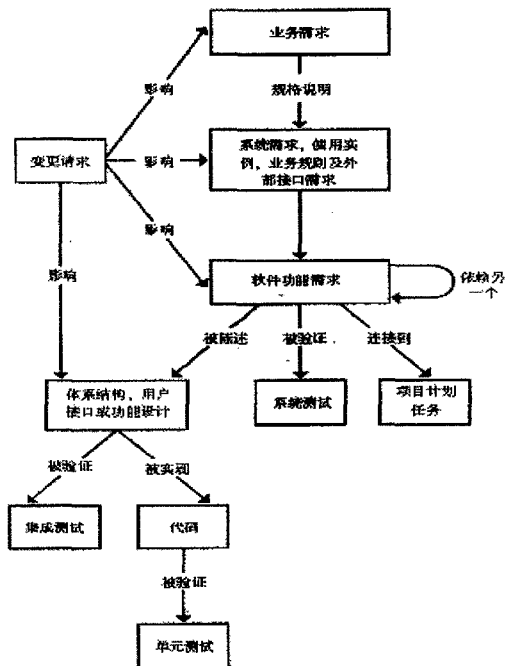


图 6-1 一些可能的需求跟踪能力联系链

IEEE 提供了跟踪能力的两个定义：

- “开发过程的两个或多个产品之间能够建立关系的程度，尤其是那些有前任及后继或者有主从关系的产品。例如，某给定软件组件的需求和设计的匹配程度。”
- “软件开发产品中每个元素能够建立其存在理由的程度；例如，数据流图中的每个元素定位它所满足需求的程度。”

上述第一个定义陈述了关于进程中一个元素与前任和后继元素之间的关系。第二个定义则是为了确定没有多余的元素存在（需求、设计等等）。

需求跟踪能力(traceability)使你能跟踪一个需求使用期限的全过程，即从需求源到实现的前后生存期。需求跟踪能力是优秀需求规格说明书的一个特征。为了实现可跟踪能力，必须统一地标识出每一个需求，以便能明确地进行查阅。

6.1.1 需求跟踪的动机

需求跟踪是需求管理的一项重要内容。需求跟踪的主要意义在于获得需求目前的实现状态，确保用户所有的需求都得到满足。可靠的跟踪能力信息可为用户在需求变更、系统维护、关键成员离开、系统再设计和类似系统设计等很多方面为组织提供参考和指导，并可以减少风险和提高项目成功的可能性。

需求跟踪可以改善产品质量，降低维护成本，而且很容易实现重用。使用需求跟踪的一些目的如下：

- 审核：跟踪能力信息可以帮助审核确保所有需求被应用。
- 变更影响分析：跟踪能力信息在增、删、改需求时可以确保不忽略每个受到影响的系统元素。
- 维护：可靠的跟踪能力信息使得维护时能正确、完整地实施变更，从而提高生产率。要是一下子不能为整个系统建立跟踪能力信息，一次可以只建立一部分，再逐渐增加，从系统的一部分着手建立，先列表需求，然后记录跟踪能力链，再逐渐拓展。
- 项目跟踪：在开发中，认真记录跟踪能力数据，就可以获得计划功能当前实现状态的记录。还未出现的联系链意味着没有相应的产品部件。
- 再设计（重新建造）：你可以列出传统系统中将要替换的功能，记录它们在新系统的需求和软件组件中的位置。通过定义跟踪能力信息链提供一种方法收集从一个现成系统的反向工程中所学到的方法。

- 重复利用：跟踪信息可以帮助你在新系统中对相同的功能利用旧系统相关资源。例如：功能设计、相关需求、代码、测试等。
- 降低风险：使部件互连关系文档化可减少由于一名关键成员离开项目带来的风险。
- 测试：测试模块、需求、代码段之间的联系链可以在测试出错时指出最可能有问题的代码段。

CMM (capacity maturity model) 的第三个成熟度级别（已定义级）要求具备需求跟踪能力。已定义级中的关键过程域——软件产品工程中就包含了对需求跟踪能力的要求。软件产品工程的目的，是为了一致地执行一个妥善定义的工程过程，该过程集成了全部软件工程活动，以便有效且高效地生产正确而一致的软件产品。软件产品工程包括采用项目定义的软件过程和适当的方法及工具来实施一系列软件工程任务，以便构造与维护软件产品。

软件产品工程对需求跟踪能力的描述为：“在软件工作产品之间，维护一致性。工作产品包括软件计划，过程描述，分配需求，软件需求，软件设计，代码，测试计划，以及测试过程。”需求跟踪过程中还定义了一些关于一个组织如何处理需求跟踪的期望。

6.1.2 需求跟踪联系链

需求跟踪能力(traceability)使用户能跟踪一个需求使用期限的全过程，即从需求源到实现的前后生存期。为了实现可跟踪能力，必须统一地标识出每一个需求，以便能明确地进行查阅。

需求文档的组成元素之间的联系链可以归纳为四种类型的链：满足链、进化链、理由链和依赖链。

满足链：用来确保软件生命周期中不同阶段的成果的一致性；

进化链：指明对象的来源，用来帮助对需求的理解和对象变更行为的跟踪；

理由链：表明对象或文档进化过程中遵循的原则和根据；

依赖链：帮助管理对象间的依赖关系，通常表现为某种限制。

在客户需求、需求和工作产品之间存在四类需求跟踪能力链：追溯到需求、从需求回溯、从需求追溯和回溯到需求。客户需求可向前追溯到需求，这样就能区分出开发过程中或开发结束后由于需求变更受到影响的需求。这也确保了需求规格说明书包括所有客户需求。同样，可以从需求回溯相应的客户需求，确认每个软件需

求的源头。如果用使用实例的形式来描述客户需求，使用实例和功能性需求之间的跟踪情况用图 6-2 表示。

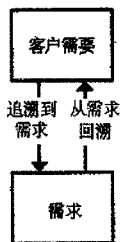


图 6-2 客户需要与需求间的跟踪联系链

由于开发过程中系统需求要转变为软件需求、设计、编码等，所以通过定义单个需求和特定的产品元素之间的(联系)链可从需求向前追溯。这种联系链使客户知道每个需求对应的产品部件，从而确保产品部件满足每个需求。而从产品部件回溯到需求联系链，使客户知道每个部件存在的原因。绝大多数项目不包括与用户需求直接相关的代码，但对于开发者却要知道为什么写这一行代码。如果不能把设计元素、代码段或测试回溯到一个需求，则表明可能有一个不必要的功能。然而，若这些孤立的元素表明了一个正当的功能，则说明需求规格说明书漏掉了一项需求。需求和工作产品间的跟踪能力联系链如图 6-3 所示。

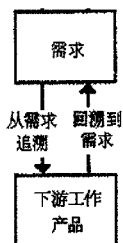


图 6-3 需求与工作产品间的跟踪联系链

6.1.3 需求跟踪能力矩阵

表示需求和别的系统元素之间的联系链的最普遍方式是使用需求跟踪能力矩阵。表 6-1 展示了一种需求跟踪能力矩阵的例子。这个表说明了每个功能性需求向后连接一个特定的使用实例，向前连接一个或多个设计、代码和测试元素。设计元素可以是模型中的对象，例如数据流图、关系数据模型中的表单、或对象类。代码参考可以是类中的方法，源代码文件名、过程或函数。加上更多的列项就可以拓展到与其它工作产品的关联，例如在线帮助文档。包括越多的细节就越花时间，但同时很容易得到相关联的软件元素，在做变更影响分析和维护时就可以节省时间。

表 6-1 一种需求跟踪能力矩阵

使用实例	功能需求量	设计元素	代码	测试实例
UC-28	catalog.query.sort	class catalog	catalog.sort()	search.7 search.8
UC-29	catalog.query.import	class catalog	catalog.import() catalog.valadiate()	search.8 search.13 search.14

跟踪能力联系链可以定义各种系统元素类型间的一对一，一对多，多对多关系。表 6-1 允许在一个表单元中填入几个元素来实现这些特征。这里是一些可能的分类：

- 一对一：一个代码模块应用一个设计元素。
- 一对多：多个测试实例验证一个功能需求。
- 多对多：每个使用实例导致多个功能性需求，而一些功能性需求常拥有几个使用实例。

创建需求跟踪能力矩阵非常有效。一旦确立使用实例基准，就准备在矩阵中添加每个使用实例演化成的功能性需求。随着软件设计、构造、测试开发的进展不断更新矩阵。例如，在实现某一功能需求后，你可以更新它在矩阵中的设计和代码单元，将需求状态设置为“已完成”。

表示跟踪能力信息的另一个方法是通过矩阵的集合，矩阵定义了系统元素对间的联系链。例如：

- 一类需求与另一类需求之间。
- 同类中不同的需求之间。
- 一类需求与测试实例之间。

可以使用这些矩阵定义需求间可能的不同联系,例如:指定/被指定、依赖于、衍生为以及限制/被限制。

表 6-2 说明了两维的跟踪能力矩阵。矩阵中绝大多数的单元是空的。每个单元指示相对应行与列之间的联系,可以使用不同的符号明确表示“追溯到”和“从…回溯”或其他联系。表 6-2 使用一个箭头表示一个功能性需求是从一个使用实例追溯来的。这些矩阵相对于表 6-1 中的单跟踪能力表更容易被机器自动支持。

表 6-2 反映使用实例与功能需求之间联系的需求跟踪能力矩阵

功能需求	使用实例			
	UC-1	UC-2	UC-3	UC-4
FR-1	←			
FR-2	←			
FR-3			←	
FR-4			←	
FR-5		←		←
FR-6			←	

6.1.4 需求跟踪能力过程

当你应用需求跟踪能力来管理工程时,可以考虑下列步骤:

- 1) 决定定义哪几种联系链,可以参见图 6-2 和 6-3 来进行。
- 2) 选择使用的跟踪能力矩阵的种类,是表 6-1 还是表 6-2。
- 3) 确定对产品哪部分维护跟踪能力信息。由关键的核心功能、高风险部分或将来维护量大的部分开始做起。
- 4) 通过修订过程和核对表来提醒开发者在需求完成或变更时更新联系链。
- 5) 制定标记性的规范,用以统一标识所有的系统元素,达到可以相互联系的目的。若必要,作文字记录,这样就可以分析系统文件,便于重建或更新跟踪能力矩阵。
- 6) 确定提供每类联系链信息的个人。
- 7) 培训项目组成员,使其接受需求跟踪能力的概念和了解重要性、这次活动的目的、跟踪能力数据存储位置、定义联系链的技术—例如,使用需求管理工具的特点。确保与会人员明白担负的责任。
- 8) 一旦有人完成某项任务就要马上更新跟踪能力数据,即要立刻通知相关人员更新需求链上的联系链。

9) 在开发过程中周期性地更新数据, 以使跟踪信息与实际相符。要是发现跟踪能力数据没完成或不正确那就说明没有达到效果。

6.2 需求可跟踪性管理

6.2.1 需求可跟踪性管理概念

如上所述, 优秀的需求说明必须具备可跟踪性。可跟踪性是需求说明必须具备的一个重要特征。要保证需求的可跟踪性, 就必须在需求开发和管理的整个过程中建立并维护不同需求之间以及需求与其它系统元素之间的联系链。这就是需求的可跟踪性管理, 即在项目开发过程中建立、检查、更新和维护不同需求之间以及需求与其它系统元素之间的关联是否正确和完整, 以保证需求的可跟踪性的过程。需求可跟踪性管理其实就是维护和管理需求跟踪联系链, 进而维护和管理需求跟踪能力矩阵的过程。

需求可跟踪性管理是贯穿于整个项目开发过程的, 包括需求开发和管理的整个过程。每当完成项目的某个阶段后, 都必须进行需求可跟踪性检查, 以保证项目元素之间联系链的正确性和完整性。正确性意味着联系链在真正具有关系的项目元素之间创建, 没有冗余的联系链, 而完整性意味着没有被遗漏的联系链。只有当完成了这项工作后, 才能进入项目的下一阶段。例如, 在需求细化阶段, 当完成第 N 级的需求定义之后, 必须进行需求可跟踪性管理, 在保证了需求联系链的正确性和完整性后, 才能进入第 $N+1$ 级的需求定义。

6.2.2 需求可跟踪性管理报告

如上所述, 可跟踪性是用来表征产品元素之间建立关系的程度。那么, 用什么指标来如何来表征这个程度呢? 需求可跟踪性管理的结果是什么? 答案是需求可跟踪性管理报告。这个报告将包含表示项目各个阶段的需求可跟踪性的指标数据。

目前, 对于如何建立跟踪关系, 如何表示这种关系, 有什么应用规则, 以及它的属性是什么等问题都没有确切的标准。项目元素之间的联系链可以有多种类型, 例如, “被实现”、“是……的一部分”、“来源于”等等。因此, 项目团队往往将根据项目过程定义、使用的特殊工具和具体程序来制定需求可跟踪性管理指标和报告模板, 以支持可跟踪性。

如第三章所述, DOORS 为用户提供了无限制关系的、多级的、用户可自定制的跟踪能力。DOORS 使用链接(Link)来表示需求之间及需求与其它项目元素之间的

联系链。DOORS 还提供了一个工具用来显示当前模块与其它模块之间的链接情况。这个工具叫 Traceability Explorer，如图 6-4 所示。

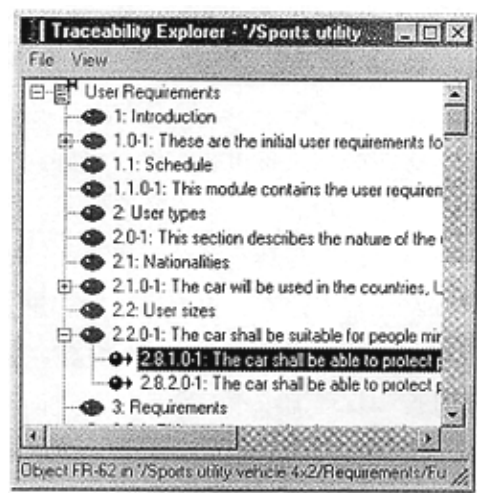


图 6-4 Traceability Explorer

在上图中，有链接的需求对象前面有符号“+”进行标记。图中选中的对象具有两个出链接。显然，这个工具虽然提供了一定的功能，使用户能一目了然的了解当前模块的链接情况，但是远远不能满足需求可跟踪性管理的要求。因此，项目团队必须制定出适用自己的需求可跟踪性管理报告，并借助工具实现该报告自动生成，以提高项目质量和工作效率。DOORS 的脚本编程功能为自动生成报告提供了强大的支持。

下面的部分将介绍笔者设计制定的一个需求可跟踪性管理报告模板，并介绍在 DOORS 下编写程序自动生成该需求可跟踪性报告。

表 6-3 是笔者设计制定的一个需求细化阶段的需求可跟踪性报告。该报告包含 5 项与跟踪性有关的数据指标，这些指标均采用百分比的形式。从这些百分比数据中，项目相关人员可以了解当前需求可跟踪性的状况，并以此为依据，做出相应行动。

表 6-3 需求可跟踪性报告

No.	数据项名称	描述	计算公式	备注
1	存在问题的需求百分比	表征存在突出技术问题的需求数量	$(\text{存在问题的需求数量} / \text{需求总量}) \times 100$	当完成创建当前级别需求的活动后进行计算，0 表示不存在这样的需求

表 6-3 需求可跟踪性报告（续）

No.	数据项名称	描述	计算公式	备注
2	丢失父链接的需求百分比	表征每个级别上没有出链接的需求数量(顶级需求除外)	$(\text{丢失出链接的需求数量} / \text{需求总量}) * 100$	当完成创建当前级别需求的活动后进行计算，0 表示不存在这样的需求
3	丢失强制属性值的需求百分比	表征存在强制属性没有值的需求数量	$(\text{存在强制属性无值的需求数量} / \text{需求总量}) * 100$	当完成创建当前级别需求的活动后进行计算，0 表示不存在这样的需求
4	丢失子链接的需求百分比	表征每个级别上没有入链接的需求数量(最后一级需求除外)	$(\text{丢失入链接的需求数量} / \text{需求总量}) * 100$	当完成创建当前级别需求的活动后，计算上一级需求，0 表示不存在这样的需求
5	存在多个父亲的需求百分比	表征存在多个出链接的需求数量	$(\text{存在多个出链接的需求数量} / \text{需求总量}) * 100$	当完成创建当前级别需求的活动后进行计算，0 表示不存在这样的需求

在 DOORS 中，使用如图 6-5 所示的模型来指示需求细化阶段相邻需求级别的需求对象之间的关系。

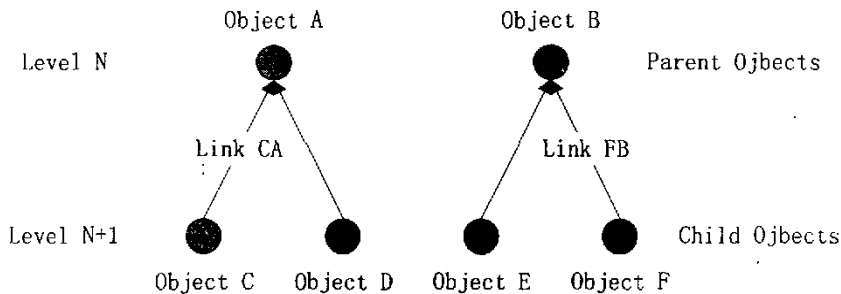


图 6-5 相邻需求级别的需求对象之间的关系

如上图所示，在完成创建 N+1 级需求的活动后，根据需求细化关系，对象 C 和对象 D 由细化对象 A 而生成，因此分别由对象 C 和对象 D 创建链接 CA 和链接 DA(图中为标出)到对象 A。对象 A 是对象 C 和 D 的父对象，对象 C 和 D 是对象 A 的子对象。链接 CA 是对象 C 的出链接，也称作父链接。同时，链接 CA 是对象 A 的入链接，也称作子链接。

对于需求属性，项目团队可以制定适合自己的需求属性集合，同时指定一些强制属性组成最小属性集合。所谓强制属性，是指在创建每一个需求时必须填写值的需求属性。制定强制需求属性是为了保证需求的信息完整性。

在自动生成该报告时，可以附加生成更详细的信息作为附录，如将那些符合统计条件的需求罗列出来，这样相关人员进行可跟踪性管理和维护就更加容易了。

图 6-6 显示了报告自动生成程序的工作流程图。

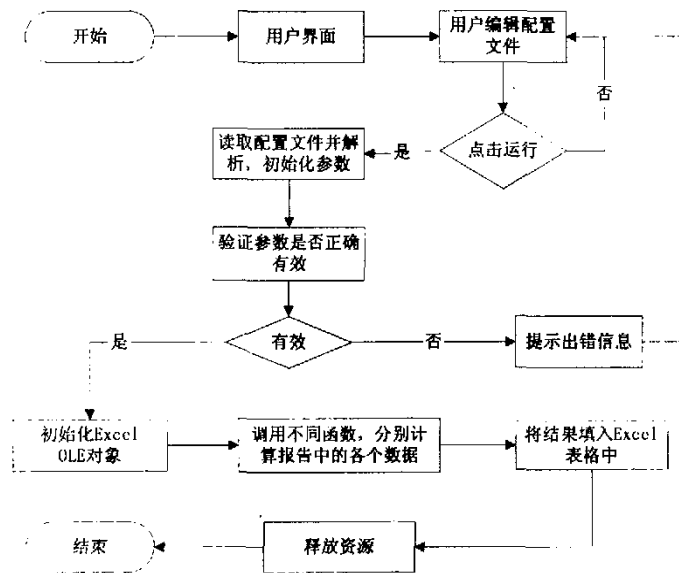


图 6-6 报告自动生成程序的工作流程图

该程序采用了配置文件的方式为用户提供方便灵活的输入参数设置。图 6-7 显示了一个配置文件的例子。该程序将运行结果直接输出到 Excel 电子表格中，用户可以直接保存生成可跟踪性报告。图 6-8 显示了一个输出到 Excel 电子表格的运行结果的例子。

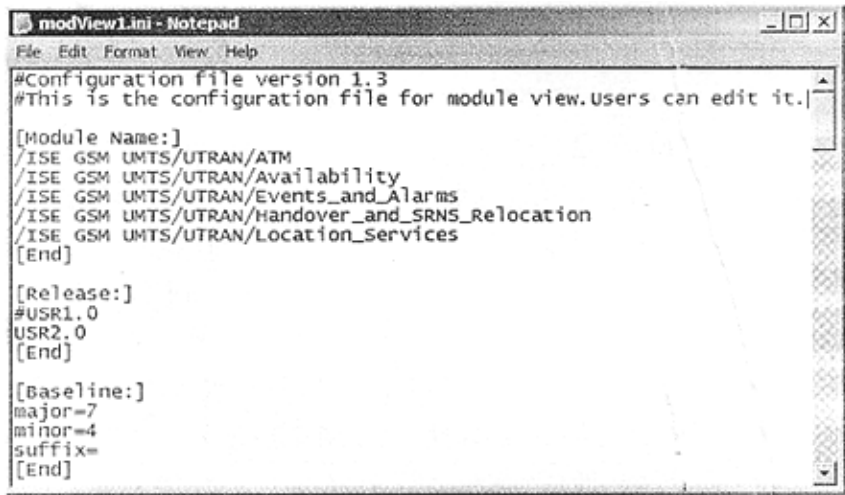
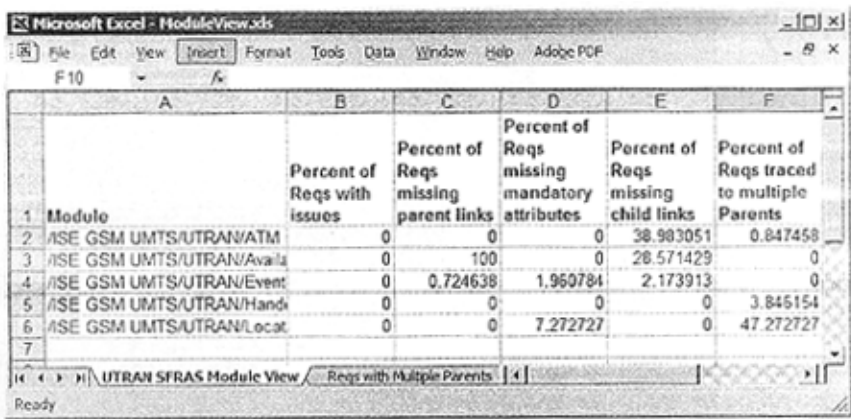


图 6-7 一个配置文件的例子



	A	B	C	D	E	F
	Module	Percent of Reqs with issues	Percent of Reqs missing parent links	Percent of Reqs missing mandatory attributes	Percent of Reqs missing child links	Percent of Reqs traced to multiple Parents
2	/ISE GSM UMTS/UTRAN/ATM	0	0	0	38.983051	0.847458
3	/ISE GSM UMTS/UTRAN/Availa	0	100	0	28.571429	0
4	/ISE GSM UMTS/UTRAN/Event	0	0.724638	1.969784	2.173913	0
5	/ISE GSM UMTS/UTRAN/Hand	0	0	0	0	3.845154
6	/ISE GSM UMTS/UTRAN/Locat	0	0	7.272727	0	47.272727
7						

图 6-8 一个输出到 Excel 电子表格的运行结果的例子

由图 6-8 可以看到，在自动生成该报告时，可以附加生成更详细的信息作为附录，图中的表单“Reqs with Multiple Parents”就是包含具有多个父亲的需求对象的详细信息，这样相关人员进行可跟踪性管理和维护就更加容易了。

第七章 结束语

在长期的软件工程实践中,人们逐步认识到,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。需求对于一个项目的重要性由此可见一斑。随着对需求研究的不断深入,学术界形成了软件工程的一个新的子领域——需求工程。需求工程可以分为需求开发和需求管理。人们对需求管理的研究已逐渐成熟。过程能力成熟度模型(CMM)对需求管理是一个有用的指导。ISO9001 在 4.3 合同评审中隐含了企业在实施软件开发过程中对需求管理的要求。

本文的研究重点是需求管理过程改进。虽然人们在需求工程和需求管理上的研究已经取得了很大的进展,但是与实际应用的距离还比较遥远。人们在需求管理实践中,必须根据自身的实际情况,结合选用的需求工具,制定符合自身需要的需求管理过程。同时,组织的软件过程应该是一个不断改进的过程。因此,本论文以笔者在某大型外企参与需求管理工作的实践为基础,对需求管理过程及其改进进行了一定的理论和实践探讨。本文分析总结了需求本身固有的特性,提出了需求细化和需求级别的概念;讨论了建立需求数据库的基本原则——层次结构统一化,并提出了一个 DOORS 需求数据库层次结构及元素命名规范;分析了现有的需求变更管理技术和过程,指出传统单级变更控制过程的缺点和不足,提出新的需求变更控制过程——两阶段需求变更控制过程,并阐述了在 DOORS 下如何实现这个过程;提出需求可跟踪性管理概念,设计需求可跟踪性管理报告模板,并在 DOORS 下编写程序自动生成该需求可跟踪性报告。

目前市场上有许多商业需求管理工具,主要有:Caliber-RM、Telelogic DOORS、QSSRequireIT、RequisitePro、RTM Workshop、Vital Link 等。本文着重介绍了 Telelogic DOORS。DOORS 是全球领先的需求管理工具,目前在全球已有超过 50,000 个用户和 1,000 多家公司在使用。本文介绍了 DOORS 的功能特性、相关概念和强大的脚本编程语言 DXL,并把创建需求数据库的统一化规范、需求细化、两阶段需求变更控制过程和需求可跟踪性管理在 DOORS 下——进行了实现和验证。

参考文献

- 【1】 Karl E.Wiegers 著, 陆丽娜、王忠民、王志敏等译, 《软件需求》, 机械工业出版社, 2000 年 7 月
- 【2】 Dean Leffingwell、Don Widrig 著, 蒋慧译, 《软件需求管理用例方法(第二版)》, 中国电力出版社, 2004 年 5 月
- 【3】 Roger S.Pressman 著, 梅宏译, 《软件工程——实践者的研究方法(第 5 版)》, 机械工业出版社, 2002 年 9 月
- 【4】 周之英著, 《现代软件工程》, 科学出版社, 2001
- 【5】 (美) 卡耐基梅隆大学软件工程研究所著, 刘孟仁等译, 《能力成熟度模型(CMM): 软件过程改进指南》, 电子工业出版社, 2001 年 7 月
- 【6】 雷辉、李怀璋、王青, 基于 ISO9001 和 CMM 的软件需求管理的研究, 计算机科学 Vol.29 No.2 2002 P87-90
- 【7】 赵晶华、苏秦, 软件需求变更过程质量管理及控制初探, 计算机应用研究 No.7 2004 P31-34
- 【8】 杨根兴、金荣得、宗宇伟, 软件需求的不确定性与解决途径, 计算机应用与软件, Vol.19 No.4 2002 P17-19
- 【9】 甘早斌、陈传波、裴先登, 基于 Web 的软件需求管理系统研究, 计算机应用研究 No.9 2003 P53-55
- 【10】 Steve Andriole, The Politics of Requirements Management, IEEE Software, Nov/Dec 1998, P82-84
- 【11】 Qing Wang, Xufang Lai, Requirements management for the incremental development model, IEEE Software, 2001, P295-301
- 【12】 Annabella Loconsole, Empirical studies on requirement management measures. IEEE, Proceedings of the 26th International Conference on Software Engineering (ICSE'04) 2004
- 【13】 Atsushi Kobayashi, Mamoru Maekawa, Need-based requirements change management, IEEE, 2001, P171-178
- 【14】 Zeljka Pozgaj, Hrvoje Sertic, Marija Boban, Effective requirement specification as a precondition for successful software development project, 25th Int.Conf. Information Technology Interfaces(ITI), June 2003, P669-674

- 【15】马捷, 需求管理和需求跟踪的策略探讨, 学位论文, 武汉大学, 2002-05
- 【16】罗俊伟, 软件需求管理工具的研究与实现, 学位论文, 中国科学院软件研究所, 2003-07
- 【17】刘小波, 软件开发中需求管理方法的研究与设计, 学位论文, 华南理工大学, 2002-03
- 【18】胡斌, 变更信息管理机制及其在 PDM 变更管理中的应用, 学位论文, 浙江大学, 2004-03
- 【19】刘丽, 软件开发工程中的需求管理探究, 学位论文, 中央财经大学, 2004-04
- 【20】Richard Stevens & James Martin, Telelogic, 什么是需求管理, 2004
- 【21】IBM Rational 需求管理技术白皮书 Ver1.0, IBM Rational Software, 2002
- 【22】DXL Reference Manual& Getting Started with DOORS, Telelogic, 2004
- 【23】武玫, 基于本体的软件需求知识管理及其系统研究, 学位论文, 同济大学, 2004-03
- 【24】贾宇清, 软件产品的客户需求管理, 学位论文, 北京航空航天大学, 2004-02
- 【25】卢梅、李明树, 软件需求工程——方法及工具评述, 计算机研究与发展, Vol.36 No.11 1999
- 【26】雍信阳、施伯乐, 需求跟踪与平滑过渡, 计算机应用, Vol.17 No.3 1997
- 【27】张家重、吕建等, 软件需求工程述评, 计算机科学, Vol.22 No.3 1995, P1289-1300
- 【28】张家重、徐家福, 需求工程研究新进展, 计算机研究与发展, Vol.35 No.1 1998, P1-5
- 【29】金芝, 没有需求就没有软件, <http://www.uml.org.cn/RequirementProject/yhxq.htm>
- 【30】Catherine Connor & Leonard Callejo, 为开发人员提供的需求管理实践, IBM, <http://www.uml.org.cn/RequirementProject/2004110401.htm>

致谢

衷心感谢我的恩师宋茂强教授！

三年来一直都在恩师的关怀、指导下学习、工作，深深体会到恩师严谨的治学态度，精深的学术造诣，虚怀若谷的风度，平易近人的性格，这一切都是我学习的榜样，在我今后的人生旅途上将一直指引着我前进。

在论文的准备、撰写过程中，我得到了陈莉萍老师和赵方老师的大力支持和帮助。她们给我提出了许多宝贵的意见，也谨在此表示最诚挚的谢意。

此外，感谢阮祖望博士及杨斌先生，他们为我提供了良好的实践机会，对我的实习工作给予了极大的帮助和指导。同时，我要感谢我的搭档彭胜任同学，我们一起完成了许多工作，他的工作给我留下了深刻的印象，并给了我很多启发和帮助。

最后，向所有关心我、帮助我的老师、同学、朋友们表示由衷谢意！