

系统工程 流程



Telelogic

需求工程



- ▲ 总结了需求工程实践的最新技术
- ▲ 讲述了适合多种系统开发背景的灵活过程
- ▲ 介绍了丰富可跟踪性的重要新概念

[美] M. Elizabeth C. Hull , Ken Jackson , A. Jeremy J. Dick 著

韩柯译

Telelogic 中国公司 审校



清华大学出版社

需求工程

[美] M. Elizabeth C. Hull, Ken Jackson, A. Jeremy J. Dick 著

韩柯译

Telelogic 中国公司 审校

清华大学出版社

北京

北京市版权局著作权合同登记号：图字 01-2003-4520 号

内 容 简 介

本书主要内容如下：解释系统工程的重要性，创建针对问题的有效解决方案；介绍系统建模使用的内部表示，包括数据流图、状态图、面向对象的方法；介绍一种通用多层需求过程；讨论有效需求管理的关键要素；概括介绍推动需求管理过程的软件工具 DOORS。

本书可适用于渴望丰富需求工程知识的读者，既包括实际工作者，也包括在校学生。

本书运用最新研究成果，由业界实践经验驱动，在如何编写和结构化需求方面会对实际工作者带来很大的启发。

Translation from the English language edition:

M.Elizabeth C.Hull, Ken Jackson, A. Jeremy J.Dick

Requirments Engineering

Copyright ©Springer-Verlag London Limited 2002

All rights reserved.

Chinese language edition published by Tsinghua University Press.

本书英文版由 Springer-Verlag 于 2002 年出版，版权为 Springer 所有。本书中文版由 Springer 授予清华大学出版社出版。未经出版者书面允许，不得以任何方式复制或抄袭本书内容。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目 (CIP) 数据

需求工程/ (美) 赫尔, (美) 杰克逊, (美) 迪克著; 韩柯译. —北京: 清华大学出版社, 2003
ISBN 7-302-07178-0

I. 需… II. ① 赫… ② 杰… ③ 迪… ④ 韩… III. 软件工程 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2003) 第 077243 号

出版者：清华大学出版社

<http://www.tup.com.cn>

社总机：010-62770175

地 址：北京清华大学学研大厦

邮 编：100084

客户服务：010-62776969

组稿编辑：曾 刚

文稿编辑：庚 永

封面设计：钱 诚

版式设计：冯彩茹

印刷者：北京国马印刷厂

装订者：三河市金元装订厂

发行者：新华书店总店北京发行所

开 本：185×260 印张：10.5 字数：224 千字

版 次：2003 年 12 月第 1 版 2003 年 12 月第 1 次印刷

书 号：ISBN 7-302-07178-0/TP·5230

印 数：1~4000

定 价：19.00 元

前 言

虽然需求工程是常识，但是感觉起来很难，而且不容易理解。正是由于这些原因，使之实施得不是非常好。机构日益增长的压力，常常变成不引入更严格需求工程方法的一个主要理由。但是，正是这些压力才使得需求工程师在帮助机构正确开发上具有更重要的作用。

系统工程对于当今业界是至关重要的，而需求工程又是整个过程中的一个重要方面。好的过程是需求工程的关键——它能够决定如何高效、快速地生产产品。在把“产品上市时间”和“满足用户需求”作为关键成功因素的全球市场竞争中，好的过程尤其重要。

需求工程也有管理的内容，因此与需求和管理有关的问题往往被结合起来，以说明怎样利用需求来管理系统开发。

本书着重介绍需求的工程化方法以及怎样帮助系统工程师创建更好的需求。本书展示了一种通用过程，帮助读者更好地理解需求工程的本质。接下来，针对开发领域和解决方案领域，对这种通用过程进行了实例化。本书也讨论了系统建模概念，并介绍了被广泛使用的各种技术和方法。本书的一个重要特色是探讨了可跟踪性的建立，以及捕获可跟踪性的方法和能够根据可跟踪性导出的指标。最后，本书概要地介绍了 DOORS 这种需求管理工具，并通过案例研究来说明本书所介绍的过程以及 DOORS 工具的功能。

本书适合业界的系统和需求工程师阅读。作为实践者，这些读者迫切需要使用需求工程来开发系统的知识。本书适合用作计算机科学、软件工程和系统工程等高年级本科生的需求工程教材，也适合计算机科学和其他工程类的研究生阅读。

本书介绍的方法以需求工程的最新研究成果为基础，不仅吸收了学术界的观点，也在很大程度上建立于业界实际工作经验基础之上，会帮助系统工程师能够更成功地管理需求（和项目）。本书是当今需求工程高速发展领域中最佳实践的一个缩影。

M. Elizabeth C. Hull, Ken Jackson, A. Jeremy J. Dick

(2002年5月)

致 谢

感谢那些个人和组织以各种方式提供的帮助。

感谢Richard Stevens, 他以多年的需求管理工作经验为我们提供了大量的帮助, 并奠定了此书的基础。他是QSS公司的创始人之一, 这个公司开发了DOORS工具。

感谢Astrium的Les Oliver帮助我们开发了“达成共识”、“鉴定”与“满足”等需求工程流程的状态图。

感谢Praxis Critical System公司提出了设计依据的最初概念, 此概念最后成为本书丰富可跟踪性的内容。

感谢Keith Collyer, Jill Burnett与Telelogic公司其他同事为本书提供的思想及评审、注释、建议与鼓励。

目 录

第 1 章 引论	1
1.1 需求引论.....	1
1.2 系统工程引论.....	3
1.3 需求与质量.....	5
1.4 需求与生命周期.....	6
1.5 需求的可跟踪性.....	9
1.6 需求与建模.....	11
1.7 需求与测试.....	13
1.8 问题与解决方案领域中的需求.....	14
1.9 如何使用本书.....	15
第 2 章 需求工程的通用过程	17
2.1 引言.....	17
2.2 开发系统.....	17
2.3 通用过程背景.....	19
2.3.1 输入需求与导出需求.....	19
2.3.2 验收准则与鉴定策略.....	21
2.4 通用过程介绍.....	22
2.4.1 理想的开发.....	22
2.4.2 在变更背景下的开发.....	23
2.5 通用过程信息模型.....	24
2.5.1 信息类.....	24
2.5.2 共识状态.....	25
2.5.3 鉴定状态.....	26
2.5.4 满足状态.....	26
2.5.5 信息模型的约束.....	28
2.6 通用过程细节.....	28
2.6.1 共识过程.....	28
2.6.2 分析与建模.....	29
2.6.3 导出需求与鉴定策略.....	31
2.7 小结.....	33

第 3 章 系统建模与需求工程	35
3.1 引言	35
3.2 针对需求工程的图形表示	35
3.2.1 数据流图	35
3.2.2 实体关系图	41
3.2.3 状态转移图	41
3.2.4 状态图	42
3.2.5 面向对象的方法	43
3.3 图形表示与信息	45
3.4 方法	45
3.4.1 方法中有什么	45
3.4.2 结构化方法	46
3.4.3 面向对象的方法	50
3.4.4 形式化方法	54
3.5 小结	55
第 4 章 编写与评审需求	56
4.1 引言	56
4.2 对需求的要求	57
4.3 确定需求文档的结构	58
4.4 关键需求	59
4.5 使用属性	59
4.6 保证需求之间的一致性	61
4.7 需求的值	62
4.8 需求的语言	63
4.9 需求样板文件	64
4.10 需求的粒度	66
4.11 编写需求语句的准则	67
4.12 小结	69
第 5 章 问题领域中的需求工程	70
5.1 什么是问题领域	70
5.2 实例化通用过程	71
5.3 与客户就需求达成共识	72
5.4 分析与建模	72
5.4.1 标识 stakeholders	72
5.4.2 创建使用场景	74
5.4.3 确定系统范围	77

5.5	导出需求	77
5.5.1	定义结构	77
5.5.2	捕获需求	80
5.6	导出鉴定策略	85
5.6.1	定义验收准则	85
5.6.2	定义鉴定策略	86
5.7	小结	86
第 6 章	解决方案领域中的需求工程	88
6.1	什么是解决方案领域	88
6.2	从 stakeholder 需求到系统需求的需求工程	89
6.2.1	生成系统模型	90
6.2.2	创建系统模型以导出系统需求	90
6.2.3	银行例子	94
6.2.4	汽车例子	96
6.2.5	通过系统模型导出需求	100
6.2.6	与设计团队就系统需求达成共识	101
6.3	从系统需求到子系统的需求工程	101
6.3.1	创建系统体系结构模型	101
6.3.2	通过体系结构设计模型导出需求	102
6.4	使用设计体系结构的其他转换	103
6.5	小结	104
第 7 章	高级可跟踪性	105
7.1	引言	105
7.2	基本可跟踪性	105
7.3	满足论据	107
7.4	需求分配	110
7.5	评审可跟踪性	110
7.6	满足论据的语言	111
7.7	丰富可跟踪性分析	111
7.8	针对鉴定的丰富可跟踪性	112
7.9	实现丰富可跟踪性	112
7.9.1	单层丰富可跟踪性	112
7.9.2	多层丰富可跟踪性	112
7.10	可跟踪性指标	113
7.10.1	宽度	114
7.10.2	深度	114

7.10.3	成长度	114
7.10.4	平衡	115
7.10.5	潜在变更	116
7.11	小结	118
第 8 章	需求工程的管理问题	119
8.1	管理简介	119
8.2	需求管理问题	120
8.3	在采办机构中管理需求	121
8.3.1	计划	121
8.3.2	监视	123
8.3.3	变更	123
8.4	供应商机构	125
8.4.1	竞标管理	125
8.4.2	开发	128
8.5	产品机构	130
8.5.1	计划	130
8.5.2	监视	133
8.5.3	变更	133
8.6	小结	134
8.6.1	计划	134
8.6.2	监视	135
8.6.3	变更	135
第 9 章	DOORS: 用于管理需求的工具	136
9.1	引言	136
9.2	需求管理案例	136
9.3	DOORS 的体系结构	137
9.4	项目、模块与对象	137
9.4.1	DOORS 数据库窗口	137
9.4.2	正式模块	138
9.4.3	对象	139
9.4.4	图片	142
9.4.5	表格	143
9.5	历史与版本控制	143
9.5.1	历史	143
9.5.2	确定基线	143
9.6	属性与视图	144

9.6.1 属性	144
9.6.2 视图	145
9.7 可跟踪性	145
9.7.1 链	145
9.7.2 可跟踪性报告	145
9.8 输入与输出	147
9.9 小结	149
参考文献	150

第 1 章 引 论

对不知道航行目的地的人来说，没有顺风。

（哲学家Lucius Annaeus Seneca，公元3—65年）

1.1 需求引论

如果系统开发项目曾经需要过“顺风”的话，那么他们现在一定更需要。高速变化的技术和日益激烈的竞争为开发过程带来与日俱增的压力。有效的需求工程是机构的核心力量，能够指导机构把握航行的方向并顺应大潮起伏的节奏。

软件是目前新产品变化的决定性力量，这是由以下 3 个关键因素决定的：

（1）任意的复杂性。最复杂的系统一般都是含有软件的系统，并且软件常常深深地嵌入在系统的组件中。这种产品的复杂性只会受到产品设计者想象力的制约。

（2）即时发布的特点。今天，公司可以构思一种新产品，用软件来实现并迅速发布到全世界。

例如，汽车制造商可以改进其诊断系统中的软件，然后用一天的时间，就可以通过电子媒介传到世界各地数以万计的所有汽车销售商中。

（3）“商品化”的组件。现在的系统是通过购入技术和现成组件构建的，产品开发周期被大大压缩。

总的来说，这些发展趋势使竞争骤然加剧，也使得在不需要大型工厂的情况下，新技术的优势可以快速转化为垄断利润。这种趋势的结果使得缩短开发周期和调动技术的时间压力大大增加。但是，仅仅缩短“产品上市市场时间”是不够的，真正的目标是缩短“将恰当的产品投放到市场的时间”。建立需求使我们能够就“恰当的产品”达成一致，并直观地看到这种“恰当的产品”。系统工程过程的一个关键部分，就是通过需求工程首先定义问题范围，然后将所有后续开发信息与问题范围关联起来。只有通过这种方式，我们才能控制和把握项目活动，开发出既适用又经济的解决方案。

需求是所有项目的基础，要根据需求定义潜在的新系统中所有 stakeholders（利益相关人）——用户、客户、供应商、开发商、商家——所需要的内容，定义为了满足这些需要系统必须完成的工作。为了使每一个人都了解需求，需求一般要用自然语言描述，因此存在着这样的挑战：在不借助行话或习语的情况下，完整、无歧义地捕获需要或问题。一旦相互沟通并达成一致，需求就会推动项目活动。但是，stakeholders 的需要可能是多种多样的，有些需要还可能相互矛盾。这些需要在开始时可能并没有被清楚地定义，可能会受他

们不能控制的因素制约，也可能会受随时间变化的其他目标的影响。没有相对稳定的需求基础，开发的项目就只能陷入困境。这就像既不知道目的地，也没有航海图的出海航行一样。需求既提供“航海图”，又提供朝目的地航行的掌舵方法。

需求共识可为系统开发计划及完成时进行验收提供基础。当必须综合考虑多方面的实际问题时，这种需求共识是必不可少的，当在开发过程中出现不可避免的变更要求时，它也是至关重要的。如果没有以前系统足够详细的模型，怎样评估变更的影响呢？如果要求恢复变更以前的需求又要如何应对呢？

即使定义了要解决的问题和潜在的解决方案，也必须评估如果不能提供令人满意的解决方案的风险。在缺乏令人信服的风险管理策略时，没有多少投资商或 Stakeholders 会支持这样的产品或系统开发。需求使人们可以在开发的早期就能管理风险。远在投入大量开发资源之前，就可以追踪与需求冲突的风险、评估其影响、理解减缓风险和善后计划的作用。

因此，需求是构成以下活动的基础：

- 项目规划；
- 风险管理；
- 验收测试；
- 综合平衡；
- 变更控制。

项目失败的原因和成功的因素所占比例见表 1.1 和表 1.2。

表1.1 项目失败的原因

原 因	所占比例
● 不完整的需求	13.1%
● 缺乏用户的参与	12.4%
缺乏资源	10.6%
● 不实际的预期	9.9%
没有执行层的支持	9.3%
● 需求/规格说明发生了变化	8.7%
缺少规划	8.1%
● 不再需要	7.5%

资料来源：Standish 集团 1995 和 1996 年；《科学美国人》1994 年 9 月号

项目失败的最常见原因并不是技术原因，表 1.1 列出了项目失败的主要原因。表中的数据来自 Standish 集团 1995 和 1996 年的调查，它给出了项目组陈述的项目失败的各种原因所占的比重。

标有符号●的原因与需求直接相关。

这些问题主要分为以下 3 大类：

- (1) 需求——或者是组织得很差、表述得很差、与用户的关联弱、变更太快或不必要

的变更，或者是不切实际的期望。

(2) 资源的管理问题——没有足够的资金、缺乏支持、没有合适的纪律原则和规划，其中很多问题都是由于缺乏需求控制所引起的。

(3) 政策——这是产生头两个问题的原因。

所有这些问题都可以以相当低的成本来解决。

表1.2 项目成功的因素

因 素	所占比例
● 用户的参与	15.9%
管理层支持	13.9%
● 清晰的需求描述	13.0%
合适的规划	9.6%
● 现实的预期	8.2%
较小的里程碑	7.7%
有才能的员工	7.2%
● 产权	5.3%

资料来源：Standish 集团 1995 和 1996 年；《科学美国人》1994 年 9 月号

从表 1.2 可以看出，项目成功因素与失败因素并不是一一对应的。管理层支持和合适的规划显然很重要——项目越大、开发时间越长，失败的可能性就越大（《科学美国人》1994 年 9 月号）。

本书讨论一般的需求工程方法和有针对性的需求管理。它解释了 stakeholder 需求和系统需求之间的差别，并讨论需求怎样被用于管理系统开发。本书还说明如何利用从 stakeholder 需求、通过系统需求到设计的可跟踪性来度量进展、管理变更、评估风险。本书通篇都考虑需求以及用来满足需求的设计组件的可测试性问题，以及如何形成可检验或可验证的需求。它强调了生成能够容易地集成和测试的设计必要性。

需求管理与项目管理有重要的接口，本书第 8 章“需求工程的管理问题”将讨论这方面的问题。

1.2 系统工程引论

本书不仅仅讨论软件需求问题，需求工程的原则和实践也适用于软件只占很小部分的整个系统。

例如，考虑从伦敦到格拉斯哥的西海岸干线铁路系统。对于系统的高层需求来说，可以从伦敦的 Euston 火车站到苏格兰的格拉斯哥全程开行时间应该控制在 250 分钟以内。提出这一需求需要综合考虑系统的所有主要组成部分：

- 列车及其速度；

- 铁轨及其支持高速列车行驶的能力；
- 火车站和车站员工，及列车在车站的等待时间；
- 火车司机及其控制列车的能力；
- 信号子系统；
- 列车控制和检测子系统；
- 电力传输子系统。

虽然信号和控制子系统软件对满足这些需求中具有至关重要的作用，但是不能单独交付这些软件。完整的解决方案应包括整个系统。事实上，大多数需求都是由作为系统整体行为所表现出的属性来满足的。

那么，这里的“系统”含义是什么呢？系统是：

一组组件——包括机器、软件和人员——他们以组织协同的方式达到某种所需结果——所要满足的需求。因此系统包括人员。在西海岸干线中，司机和车站员工及他们所接受的培训和所使用的过程，与软件和机器组件一样重要。

由于组件必须彼此协同，因此组件之间的接口是系统（以及需求）工程中至关重要的考虑因素，即人员和机器组件之间的接口、机器组件之间的接口、软件组件之间的接口。铁路系统中机器到机器接口的一个例子是列车轮子与铁轨的接触方式。除了物理上的考虑外（通过设计使列车能够沿铁轨行驶而不会滑出轨道），经过铁轨的电流可能被用来作为列车控制子系统的一部分来检测列车的位置。

“系统”概念的核心是“新出现的属性”的思想。这意味着系统的有用性不是取决于系统的任何具体部分，而是系统组件交互的行为所显示的新出现的属性。新出现的属性可能有利用价值，可以被设计到系统中，以使系统有用；或者新出现的属性有不期望的副作用，例如对环境的危害。系统工程的任务就是利用有价值的新出现的属性，而避免不必要的新属性。

另一个重要的概念是“系统的系统”。每个系统都可以被用于构造更大的外围系统。例如，西海岸干线是更大铁路系统的一部分，与其他干线和支线线路构成新的铁路网。整个铁路系统又是更大运输系统的一部分，与所有种类的公路和空运网络交互。运输系统本身是提供货物和人员交通的必要基础设施，也是构成国家经济系统的一部分。而国家又是世界的一部分，以此类推。

正确地理解系统的需求，就是理解系统的外围系统。系统的正确功能往往取决于外围系统所提供的条件。例如，直升飞机的飞行能力取决于地球所提供的环境，地球的重力场和空气。

再举一个非常简单的例子（图 1.1）：杯子。杯子显然有一些组件：一个杯把和一个碗型的容器。这些组件有什么用途呢？杯碗是用来盛液体的，杯把的用途是让人拿住碗，同时又不会被烫伤。我们可以由此得出杯子的用途，也就是需求，是使人能够将热的液体送到嘴里而液体不会溅出，人也不会被烫伤。但是还有其他一些因素需要考虑。

- 杯子本身是没有用的，它要依靠人胳膊的机械运动来达到其目的。
- 杯子的杯碗部分要依靠重力的存在才能发挥作用。杯子还必须被正确使用：杯子

拿反了就会使液体倒出，并可能导致烫伤。

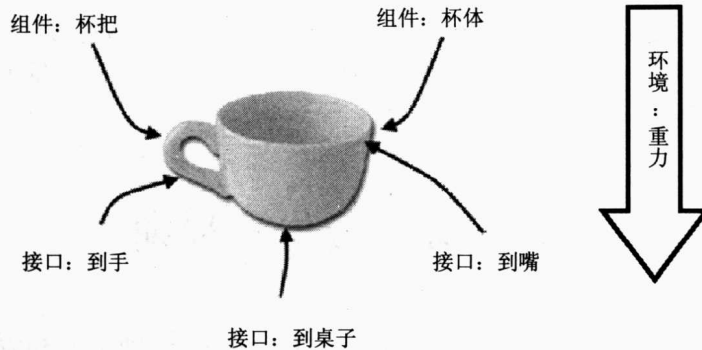


图1.1 作为非常简单系统的一个杯子

这种简单杯子要具备其用途的功能，最终取决于：

- 从组件交互所表现出的属性；
- 与外部组件的合适接口；
- 正确嵌入外围系统——由人手拿住，并用胳膊举起来；
- 适当环境的存在——在失重条件下将需要另一种解决方案。

总之，需求工程必须把系统的本质考虑进去。最重要的考虑是新出现的属性、外部环境的制约与保障、以及与周围系统的接口。

1.3 需求与质量

没有需求所造成的后果很多，并且是多种多样的。生活中有大量证据说明系统失败是因为没有合适地组织需求。不过，虽然系统初看起来可能很正常，但是如果这不是用户想要或需要的系统，那么这个系统也是没有用的。

考虑需求和质量之间的关系很有意思。术语“质量”可以用不同的方式来理解。当说到高质量的汽车时，人们可能会提到劳斯·罗伊斯、奔驰或美洲虎。这种“质量”和“豪华”两个概念的内在混淆，在为一年一度的 RAC 拉力赛选最佳赛车时就会暴露出来。不管是劳斯·罗伊斯，还是奔驰或美洲虎都不会选中，因为这些车都没有合适的重量/功率比、离地距离和稳定性等性质。最近的历史说明，在同类中具有最佳质量的汽车是一种 Skoda 车，并不是豪华汽车，但是 Skoda 具有完成其功能的真正“质量”。

质量是“对用途的适合性”，或是需求的符合性，即提供能够满足客户需要的某种功能，并保证考虑了所有 stakeholders 的需要。

在第 8 章将会介绍，需求工程是其他管理因素（例如成本和进度）的辅助，因为需求工程强调交付产品的质量。所有管理决策都是成本、进度和质量的折衷，这 3 个因素构成相互关联的 3 个轴。

由于需求工程是应用于开发生命周期初期的学科，通过适当的需求管理来对质量进行提升会产生很大的比例放大效果。开发早期阶段投入的相对很少的工作，在后期阶段能够得到几倍的回报。格言“质量是免费的”（Phil Crosby 的一本书的书名）仍然是正确的，因为开发工作开始时投入的工作，会节省本来必须要在后期投入的大量工作。改进需求意味着改进产品的质量。

1.4 需求与生命周期

人们经常错误地认为，需求工程只是在产品开发开始时需要执行和完成的一个阶段。本节的目的就是说明需求工程在开发的各个阶段都有着至关重要的作用。

让我们先考虑开发过程的最后工作：验收测试。根据什么验收系统呢？答案是 stakeholder 需求。可以看出，最初开发的需求在开发的最后阶段仍然要使用。

表明各个开发阶段的经典 V 字模型就是以测试和需求之间的关系为基础的。图 1.2 给出了开发各个阶段的这种关系。

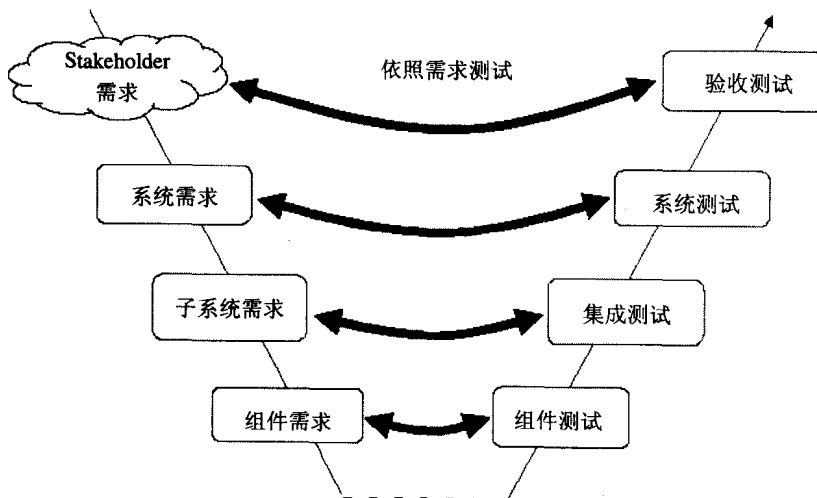


图1.2 V字模型中的需求

V 字模型还把开发分成层次，每个层次都研究适于相应开发阶段所关心的问题。尽管每个层次可以使用略有不同的过程，但是需求使用的基本模式是相同的，第 2 章在介绍一般过程时还会讨论这一点。图 1.3 给出了各个层次需求工程的主要考虑。

需求在机构中可以充当的另一种角色是充当项目之间一种沟通的手段。这是一个好想法，因为很多机构都希望：

- 尽可能扩大项目之间的产品重用；
- 管理相似产品系列；

- ❑ 通过程式化管理来协调活动;
- ❑ 通过学习其他项目的经验来优化过程。

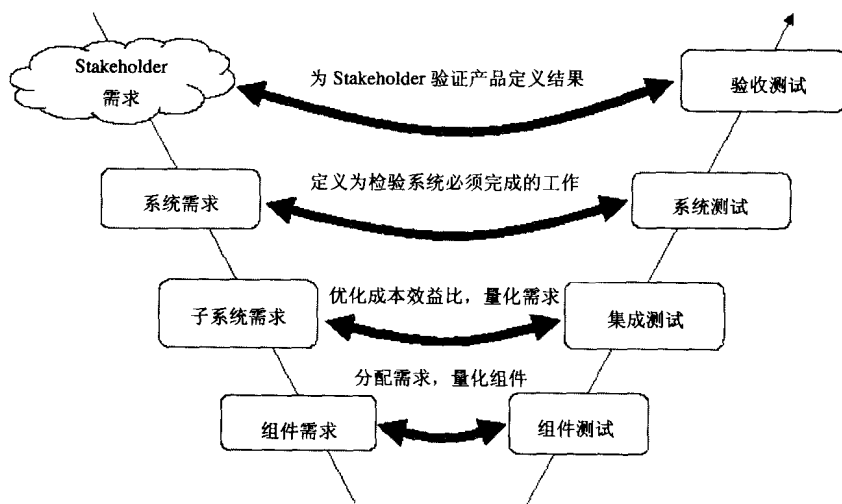


图1.3 不同层次中的需求工程

一组好的 stakeholder 需求可以为高级管理层提供所要开发内容的简捷的非技术描述。类似地，系统需求也可以对开发项目作很好的技术归纳。这些描述可以用作与其他活动进行比较的基础，如图 1.4 所示。

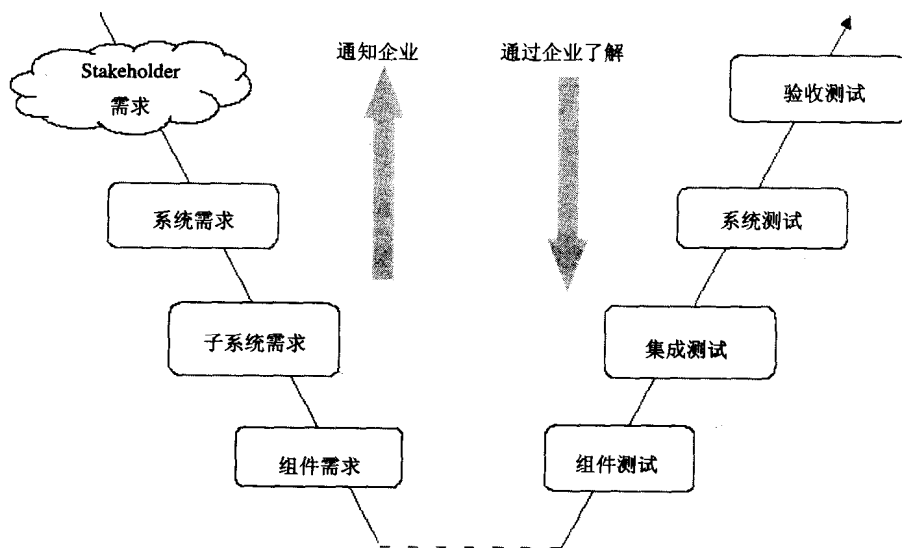


图1.4 企业需求工程

如果需求在系统开发中发挥这样的核心作用，则需求就需要被维护。在修改产品设计时，没有同时更新需求以反映设计上的变化会为开发的后续阶段积累大量问题。因此，需