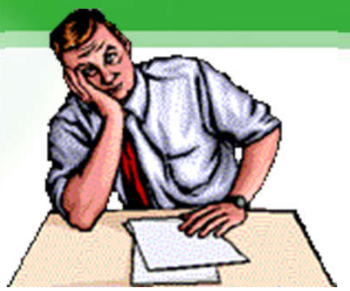
# 第5章 理解需求

王美红

# 问题……

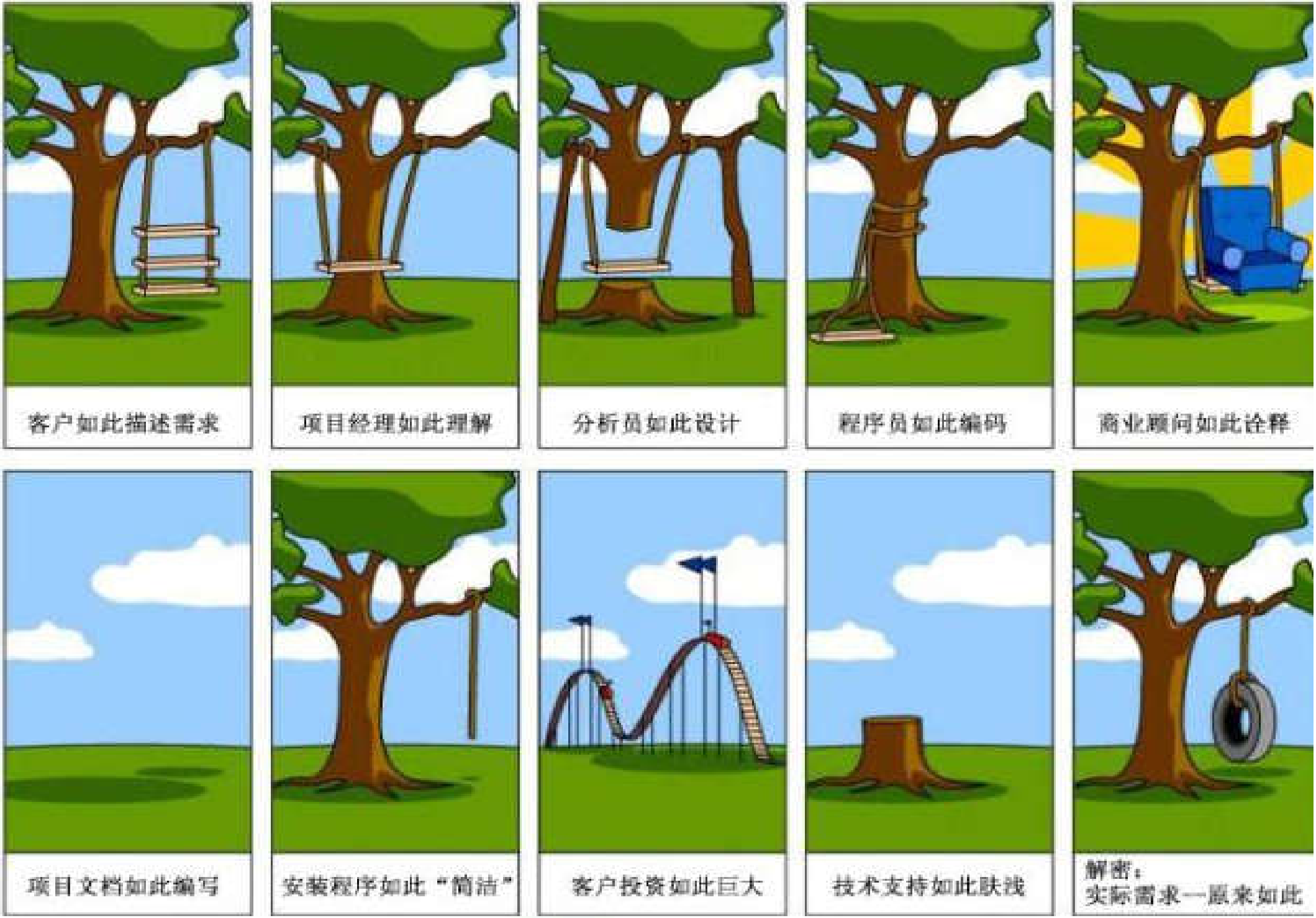


1. 用户在软件需求分析过程中重要吗?请说明理由

2. 软件需求分析是软件工程过程中交换意见最频繁的步骤,为什么交换意见的途径会经常阻塞?

# 为什么需求工程非常困难？

- 客户说不清楚需求
- 需求自身不断变动
- 分析人员或客户理解有误


客户如此描述需求
项目经理如此理解
分析员如此设计
程序员如此编码
商业顾问如此诠释
项目文档如此编写
安装程序如此“简洁”
客户投资如此巨大
技术支持如此肤浅
解密：
实际需求—原来如此

# 主要内容

- 需求工程的概念
- 需求工程的任务
- 启动需求工程过程
- 导出需求
- 开发用例

# 需求分析方法分类

- 结构化分析方法
  - 依据系统内部的逻辑关系将复杂的系统分解为易于理解和分析的子系统，是一种直接根据数据流划分功能层次的分析方法。
- 面向对象分析方法
  - 根据获得的需求信息从中抽象出类与对象并分析他们之间的静态关系，再结合实际问题，确定对象的动态行为以及对象间的信息传递，以此建立需求模型。

# 5.1 需求工程

- 需求工程（Requirement Engineering, RE）
  - 是指致力于不断理解需求的大量任务和技术。
- 需求工程在设计和构造之间建立起联系的桥梁。

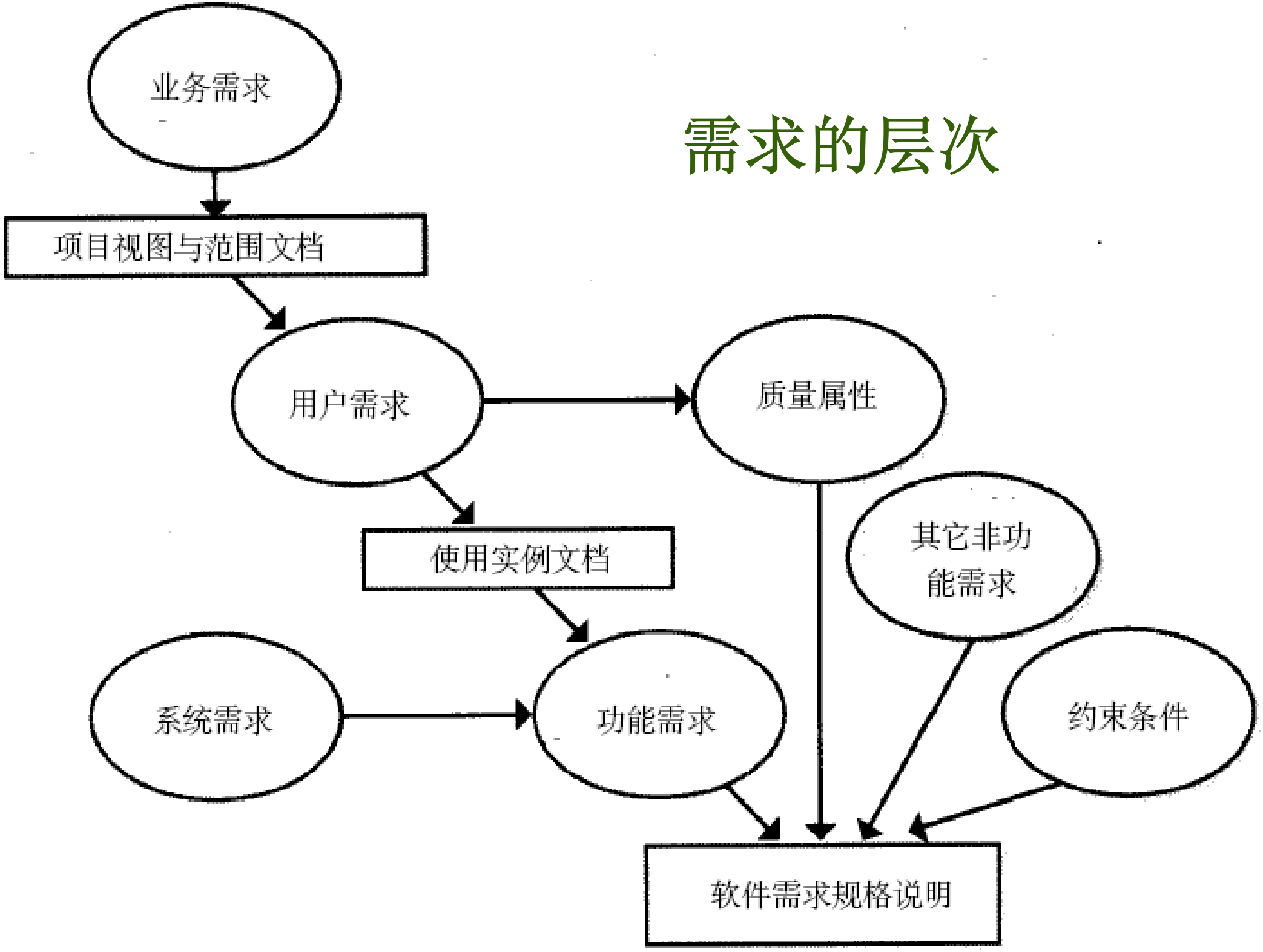
# 软件需求

- 软件需求包括三个不同的层次:
  1. 业务需求
  2. 用户需求
  3. 功能需求—也包括非功能需求。

# 需求分析的三个层次

- 业务需求：
  - 反映了组织机构或客户对系统、产品高层次的目标要求。
- 用户需求：
  - 文档描述了用户使用产品必须要完成的任务。
- 功能需求：
  - 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。

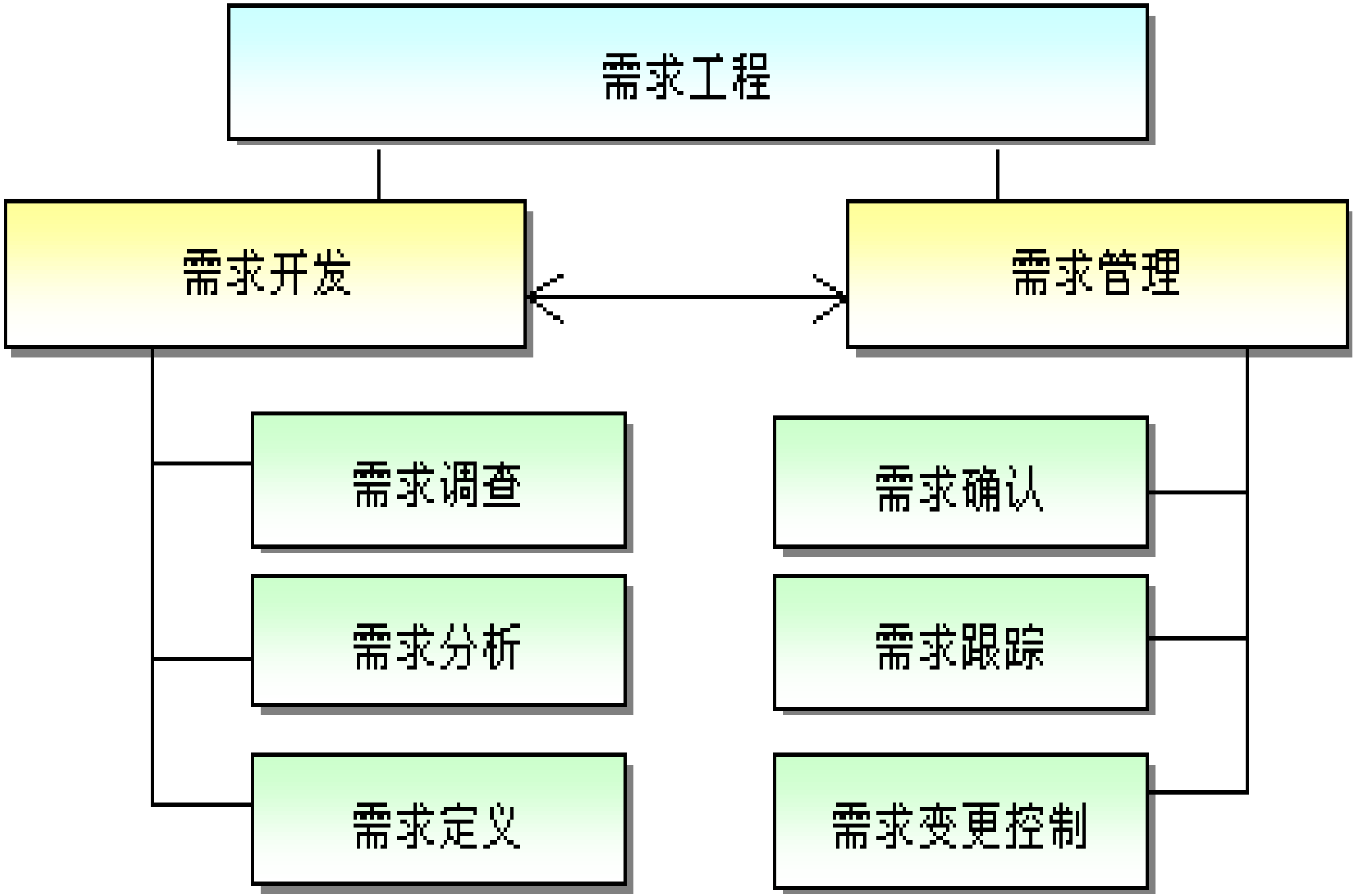
# 需求的层次


业务需求
项目视图与范围文档
用户需求
质量属性
使用实例文档
其它非功能需求
系统需求
功能需求
约束条件
软件需求规格说明

# 5.1 需求工程

- 把所有与需求直接相关的活动通称为需求工程。
- 需求工程中的活动可分为两大类:
  - 一类属于需求开发
  - 另一类属于需求管理。

# 需求工程的结构图



# 5.1 需求工程

- 需求工程通过执行七个不同的活动来完成：
  1. 起始
  2. 导出
  3. 精化
  4. 协商
  5. 规格说明
  6. 确认
  7. 管理

# 5.1 需求工程

- 1. 起始：
  - 软件工程是询问一些似乎与项目无直接关系的问题
  - 泛谈起始，有各种各样的情况
  - 目的是对问题、方案需求方、期望方案的本质、客户和开发人员之间初步的交流和合作的效果建立基本的谅解

# 5.1 需求工程

- 2. 导出
  - 询问客户、用户和其他人，系统或产品的目标是什么？想要实现什么？系统和产品任何满足业务的要求，最终系统和产品如何用于日常工作？
  - 非常困难：范围问题、理解问题、异变问题

# 5.1 需求工程

- 3. 精化/细化
  - 将起始和导出阶段获得的信息进行扩展和提炼
  - 是一个分析建模动作
  - 精化的最终结果：一个分析模型，定义了问题的信息域、功能域和行为域

# 5.1 需求工程

- 4. 协商
  - 需求工程师必须通过协商的过程调节各种冲突
    - 按优先级讨论冲突
    - 识别和分析风险
    - 粗略“估算”开发工作量，并评估每项需求对项目成本和交付时间的影响
    - 使用迭代，删除、细化或修改需求，以便各方达到一定的满意度

# 5.1 需求工程

- 5. 规格说明(specification)
  - 把前面的成果用文字或其它方式明示出来。
  - 可以是一份写好的文档，一套图形化的模型，一个形式化的数学模型，一组使用场景，一个原型或上述各项的任意组合

# 5.1 需求工程

- 6. 确认：
  - 要检查规格说明以保证：
    - 所有的系统需求已被无歧义地说明；不一致性、疏漏和错误已被检测出并被纠正；工作产品符合为过程、项目和产品建立的标准。
  - 由第三方（通常为评审组）完成

# 5.1 需求工程

- 7. 需求管理
  - 用于帮助项目组在项目进展中标识、控制和跟踪需求以及变更需求的一组活动。
  - 解决方法：特征跟踪表、来源跟踪表、依赖跟踪表、子系统跟踪表、接口跟踪表等。

# 5.2 建立根基

- 1. 确定共利益者
  - 直接或间接从正在开发的系统中获益的人
  - 比如业务操作管理人员、产品管理人员、市场营销人员、内部或外部客户、最终用户、顾问、产品工程师、软件工程师、支持和维护工程师以及其他人员

# 5.2 建立根基

- 2.识别多种观点
  - 需求工程师就是把所有共利益者提供的信息（包括不一致或者矛盾的需求）分类，分类的方法应该便于决策制定者为系统选择一个内部一致的需求集合。

# 5.2 建立根基

- 3. 协作
  - 如何协作？
  - 需求工程师的主要任务是标识公共区域和矛盾区域
  - 一个有效的方法是使用“优先点”，所有共利益者都分配一定数量的优先点

# 5.2 建立根基

- 4. 首次提问 – “与环境无关”
  - 第一组与环境无关的问题集中于客户和其他共利益者、整体目标、收益：
    - 谁是这项工作的最初提出者？
    - 谁将使用该解决方案
    - 成功的解决方案将带来什么样的经济效益？
    - 存在别的解决方法吗？
    - ……

# 5.2 建立根基

- 下列一组有软件开发组更好地理解问题，并允许客户表达其他对解决方案的看法：
  - 如何描述由某成功的解决方案产生的“良好的”输出？
  - 该解决方案强调了什么问题？
  - 能向我们展示（或描述）解决方案的使用环境吗？
  - 存在影响解决方案的特殊性能问题或约束吗？
  - ……

# 5.2 建立根基

– 最后一组问题关注与沟通活动本身的效率：

- 你是回答这些问题的最合适人选吗？你的回答是“正式的”吗？
- 你的提问和你解决的问题相关吗？
- 我的问题是否太多了？
- 还有其他人员可以提供更多的信息吗？
- 还有我应该问的其他问题吗？

# 5.2 建立根基

- 5. 非功能性需求
  - 可以将非功能性需求描述成质量属性、性能属性、安全属性或一个系统中的常规限制。
    - 利益相关者通常不能清晰地表达这些内容
    - 两阶段方法：
      - 第一阶段：建立软件工程指南和NFR(例如可用性、可测性、安全性或可维护性）列表的关系矩阵。
      - 第二阶段：根据一套决策规则（用来决定实施哪些指南，放弃哪些指南），划分出同类的非功能需求，建立优先级

# 5.2 建立根基

- 6. 可追溯性
  - 指软件工程工作产品间记录的文档化链接（例如需求和测试用例）
  - 需求工程师用可追溯矩阵表示需求和其他软件工程产品间的相互关系。

| | 工作产品 |
| --- | --- |
| 需求 | 两者之间关联 |

# 5.3 获取需求

- 导出需求的方法：
  ① 访谈
  ② 面向数据流自顶向下求精
  ③ 协同需求获取
  ④ 快速建立软件原型
  ⑤ 质量功能部署
  ⑥ 用户场景
  ⑦ ……

# 沟通的原则

1. 倾听；
2. 有准备的沟通；
3. 沟通活动要有人推动；
4. 最好当面沟通；
5. 记笔记并记录所有决定；
6. 保持通力协作；
7. 把讨论集中在限定的范围内；
8. 如果某些东西很难表达清楚，就采用图形表示
9. （1）一旦认可某件事情，转换话题；（2）如果不认可某件事情，转换话题；（3）如果某项特性、功能不清晰或当时无法澄清，转换话题；
10. 协商不是一场竞赛或一场游戏，双赢才能发挥协商的最大价值。

# 5.3 获取需求

① 访谈

- 正式访谈——事先准备好的具体问题
- 非正式访谈——自由问答

- 可借助：
  - 调查表再针对性访问
  - 情景分析技术(对用户将来使用目标系统解决某个具体问题的方法和结果进行分析)

# 某出版社系统调查表

| 编号 | 提出问题 |
| --- | --- |
| 1 | 您在哪个部门工作？ |
| 2 | 出版业务流程是什么？ |
| 3 | 您每日都处理哪些文件、数据、报表？ |
| 4 | 工作中手工处理特别麻烦的事情是什么？ |
| 5 | 工作中手工处理什么问题解决不了？影响效率的问题有哪些？ |
| 6 | 您认为提高工作效率，节省工作时间，减轻工作强度可采取哪些办法？ |

# 某出版社系统调查表

| 编号 | 提出问题 |
| --- | --- |
| 7 | 您的部门需要成本核算和统计的内容有哪些？ |
| 8 | 您的部门采用计算机管理工作情况如何？ |
| 9 | 如何改进业务流程使之更合理？ |
| 10 | 哪些问题是目前传统手工方法根本无法解决的？ |
| 11 | 出版社计算机管理信息系统需要解决什么问题？ |

# 软件需求调查表的编写方法与实例

# 5.3 获取需求

② 面向数据流自顶向下求精

- 数据决定了需要的处理和算法，是需求分析的出发点
- 需求分析的目标之一是在可行性研究得到的高层数据流图基础上，把数据流和数据存储定义到元素级
- 通过对未知数据和其需要的算法的请教，深入认识系统
- 分析员借助数据流图、数据字典和IPO图向用户解释输入数据是怎么一步步转变成输出数据的。

# 与用户沟通需求的方法

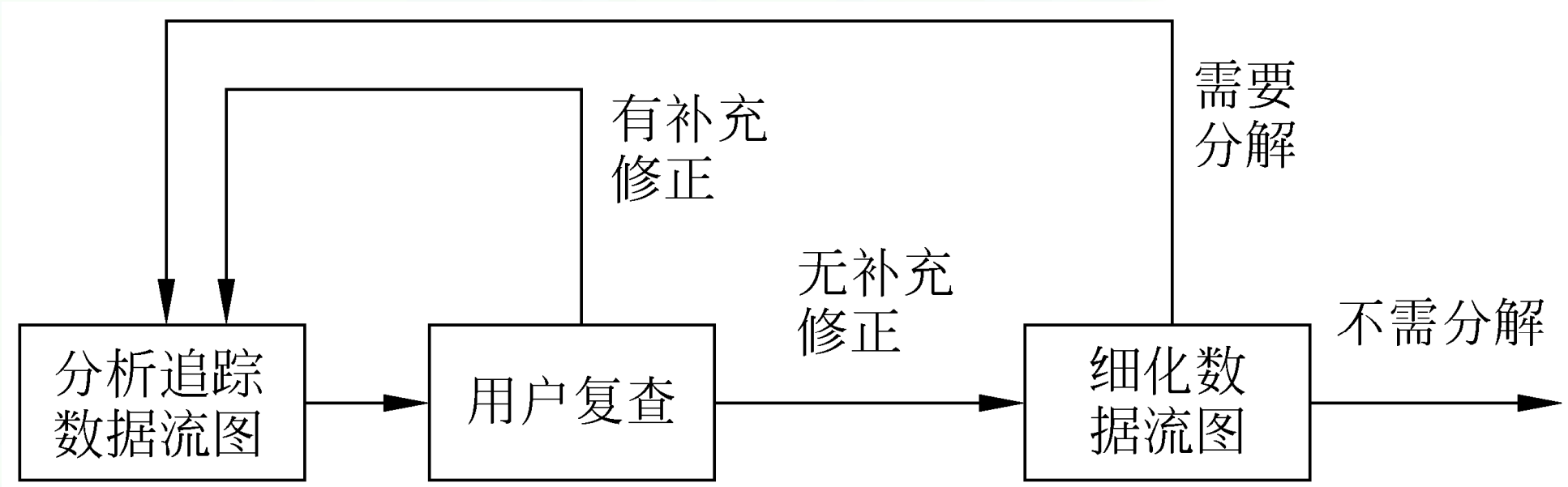
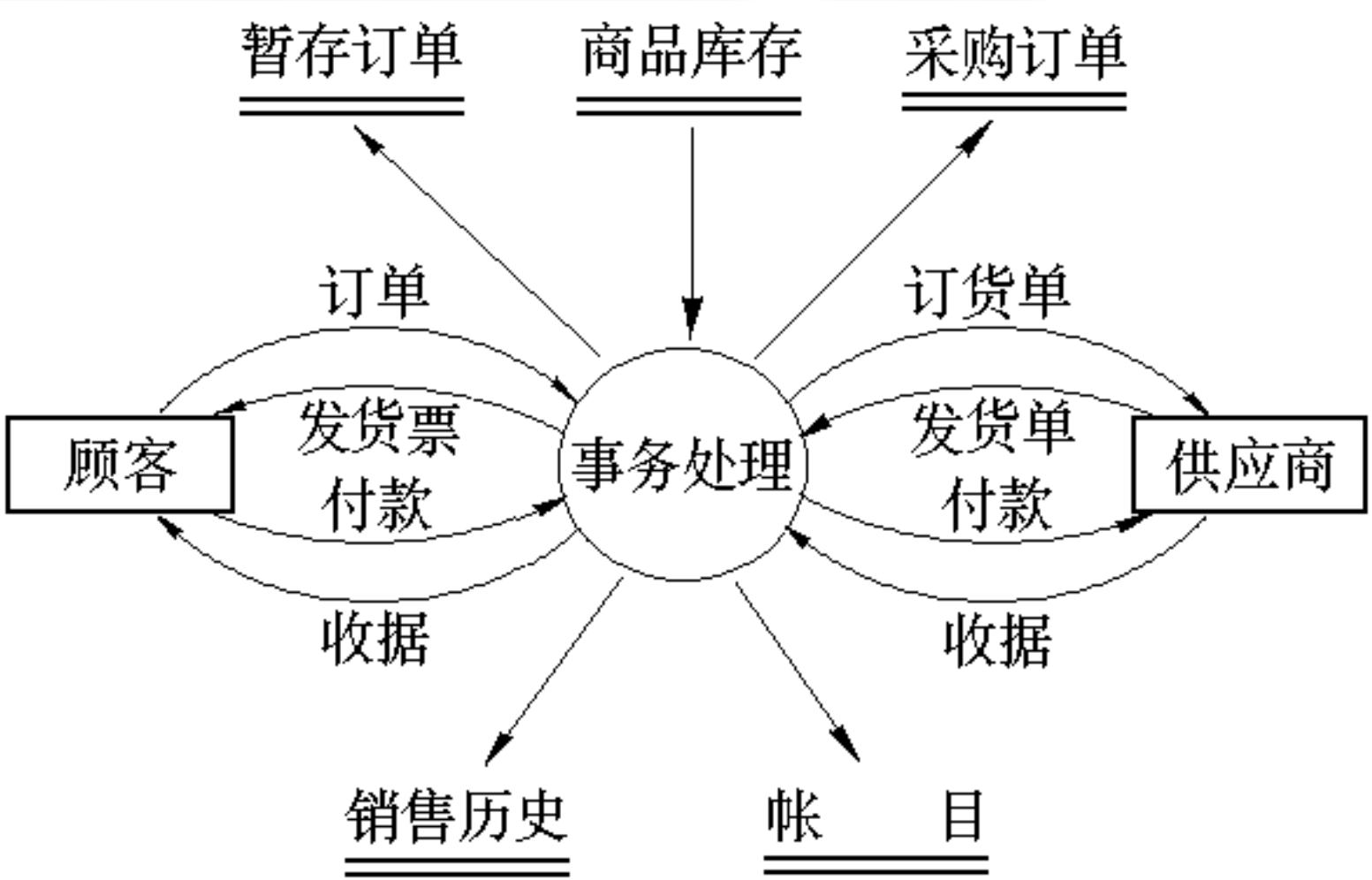


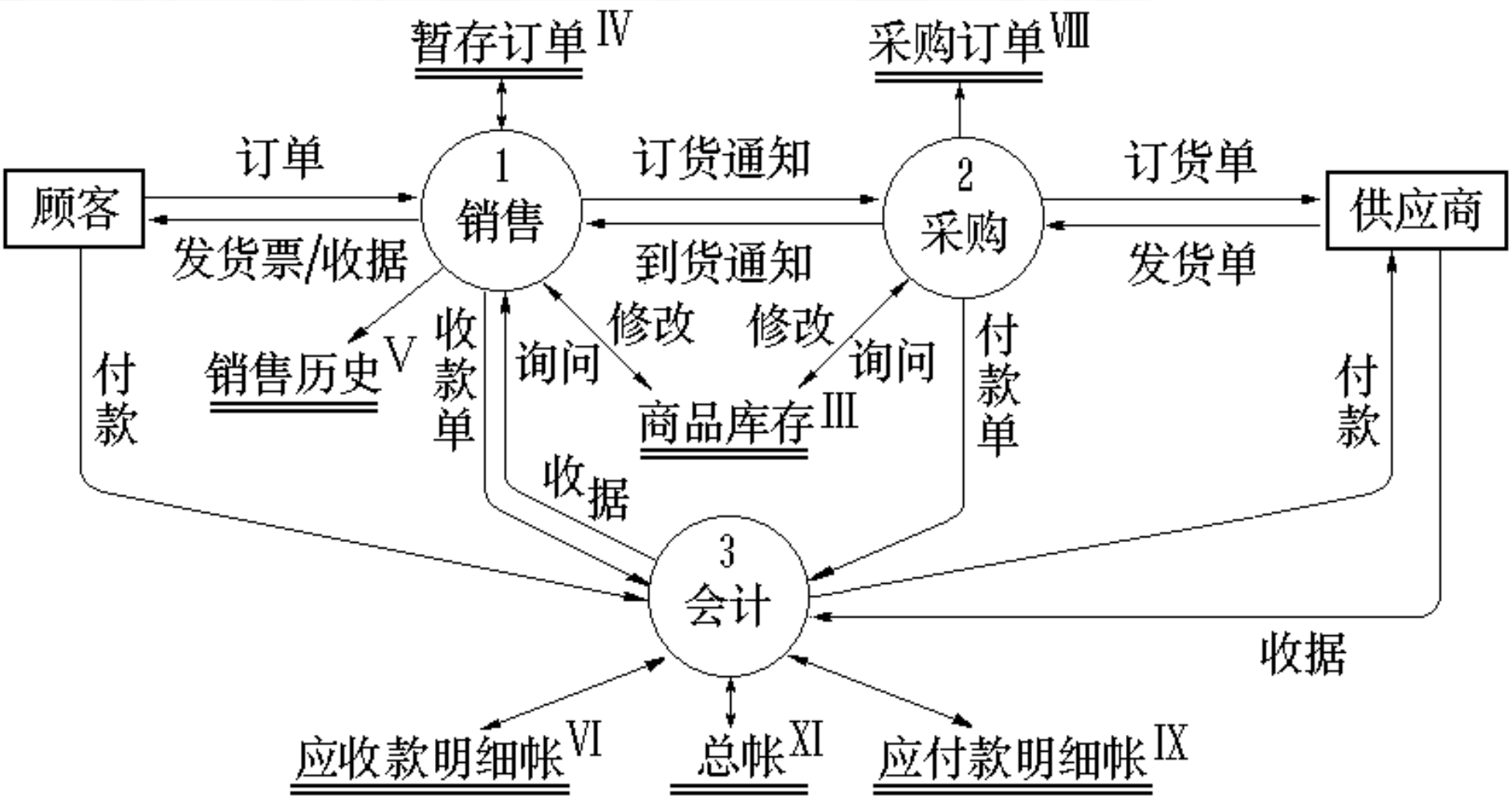

图　面向数据流自顶向下求精过程

# 数据流图举例

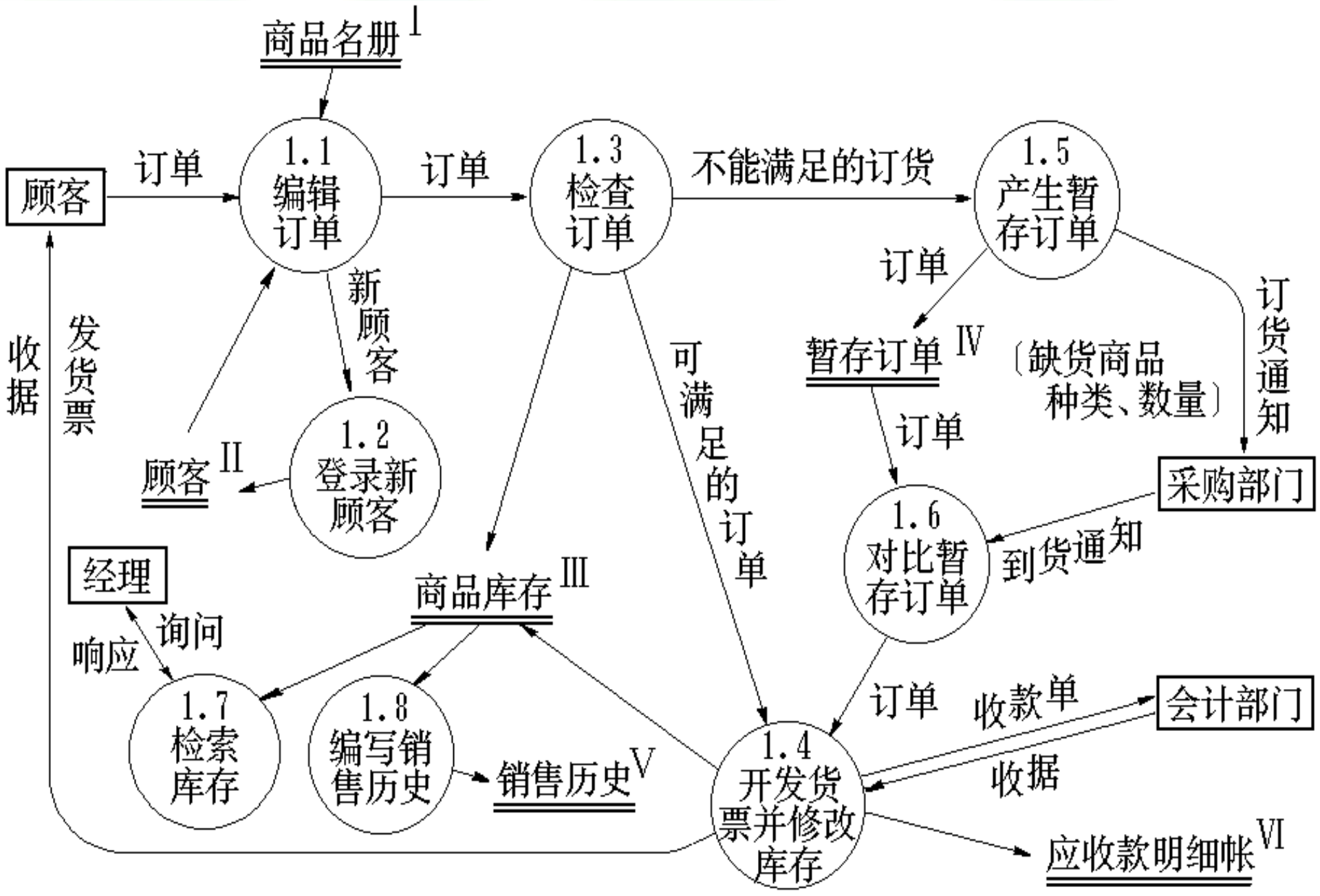
商店业务处理系统



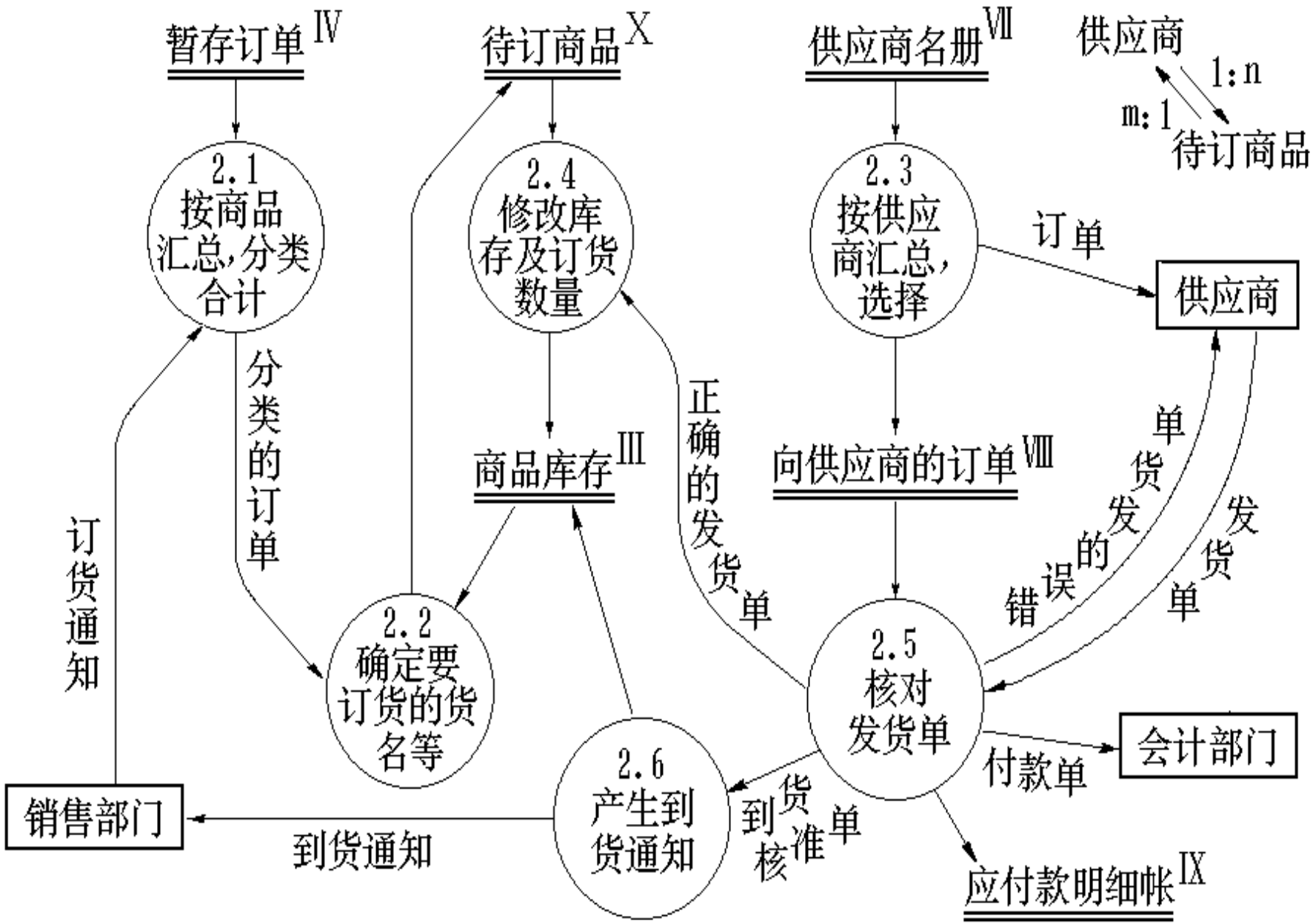
# 第一层数据流图


暂存订单 IV
采购订单 VIII
订单
订货通知
订货单
顾客
1
销售
2
采购
供应商
发货票/收据
到货通知
发货单
付款
销售历史 V
收款单
询问
修改
修改
询问
付款单
付款
商品库存 III
收据
3
会计
收据
应收款明细帐 VI
总帐 XI
应付款明细帐 IX

# 销售细化


商品名册 I
顾客
订单
1.1 编辑订单
订单
1.3 检查订单
不能满足的订货
1.5 产生暂存订单
新顾客
收据
发货票
1.2 登录新顾客
顾客 II
可满足的订单
暂存订单 IV
〔缺货商品种类、数量〕
订单
订货通知
采购部门
1.6 对比暂存订单
到货通知
经理
响应
询问
商品库存 III
1.7 检索库存
1.8 编写销售历史
销售历史 V
1.4 开发货票并修改库存
订单
收款单
会计部门
收据
应收款明细帐 VI

# 采购细化


暂存订单 Ⅳ
待订商品 X
供应商名册 Ⅶ
供应商
1:n
m:1
待订商品
2.1 按商品汇总,分类合计
2.4 修改库存及订货数量
2.3 按供应商汇总,选择
订单
供应商
分类的订单
正确的发货单
商品库存 Ⅲ
向供应商的订单 Ⅷ
单货发的误错
发货单
订货通知
2.2 确定要订货的货名等
2.5 核对发货单
2.6 产生到货通知
会计部门
付款单
销售部门
到货通知
到货核准单
应付款明细帐 Ⅸ

# 5.3 获取需求

③ 协同需求获取

- 面向团队的需求获取方法
- 共利益者和开发人员的团队共同完成：确认问题，为解决方案的要素提供建议，协商不同的方法，以及说明初步的解决方案需求集合。

# 5.3 获取需求

– 一些基础原则：

- 会议由软件工程师和客户（连同其他的共利益者）共同举办和参与
- 制定筹备和参与会议的规则。
- 建议拟定一个会议议程，这个议程既要足够正式，使其涵盖所有的重要点；但也不能太正式，以鼓励思维的自由交流

# 5.3 获取需求

- 由一个“主持人”（可以是客户、开发人员或其他外人）控制会议。
- 使用某种“定义机制”（可以是工作表、活动挂图、不干胶贴纸或电子公告牌、聊天室或虚拟论坛）
- 目的是识别问题，提出解决问题的要素，协商不同的方法以及在有利于完成目标的氛围中刻画初步解决方案的需求问题。

# 5.3 获取需求

- 简易的应用规格说明技术
  - 面向团队的需求收集法，提倡用户与开发者密切合作，共同标识问题，提出解决方案要素，商讨不同方案并指定基本需求。

# 5.3 获取需求

- 简易的应用规格说明技术典型过程：
  - 初步访谈，初步确定待解决的问题的范围和解决方案
  - 开发者和用户分别写出“产品需求”，要求每位与会者在开会的前几天认真审查产品需求，希望能准确地表达出每个人对目标系统的认识。
  - 会议上为每个议题首先创建组合列表，然后创建一张意见一致的列表(对象、服务、约束和性能)
  - 分小组，为每张列表的项目制定小型规格说明
  - 合并展示小型规格说明，并讨论。
  - 每个与会者制定一套确认标准，并提交讨论，最后创建出一件一致的确认标准。由一到多名与会者起草软件需求规格说明书。

# 5.3 获取需求

④ 快速建立软件原型

- 是最准确、最有效、最强大的需求分析技术
- 构造原型的要点是：应该事先用户能看得见的功能（如屏幕演示或打印报表），省略目标系统的“隐含”功能（例如，修改文件）
- 快速原型的特性：快速、容易修改

# 5.3 获取需求

- 快速建立软件原型可借助方法和工具：
  - 第四代技术（数据库查询和报表语言、程序和应用系统生成器及其他非常高级的非过程语言）
  - 可重用的软件构件
  - 形式化规格说明和原型环境

# 5.3 获取需求

⑤ 质量功能部署（Quality Function Development, QFD）

- 是把顾客或市场的要求转化为设计要求、零部件特性、工艺要求、生产要求的多层次演绎分析方法。
- QFD于70年代初起源于日本三菱重工的神户造船厂应付大量的资金支出和严格的政府法规，取得了很大的成功。
- 他们用矩阵的形式将顾客需求和政府法规同如何实现这些要求的控制因素联系起来。

# 5.3 获取需求

- **QFD步骤**

1. 关键顾客需求→产品特性
2. 关键产品特性→部件特性
3. 关键部件特性→过程特性
4. 过程特性→生产特性

# 5.2 建立根基

- 质量屋
  - 是QFD的核心
  - 质量屋是一种确定顾客需求和相应产品或服务性能之间联系的图示方法。
  - 各阶段质量屋不同

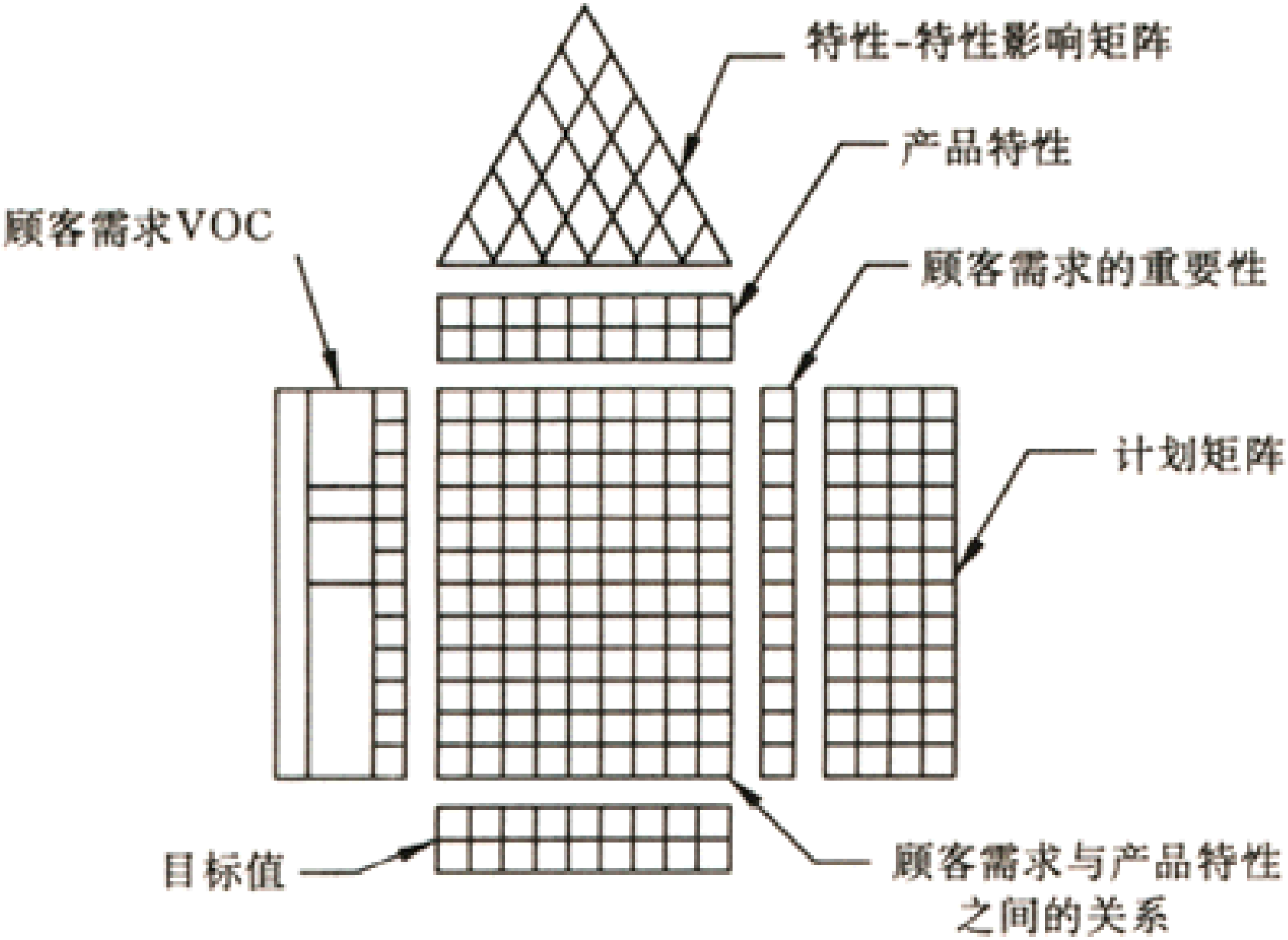
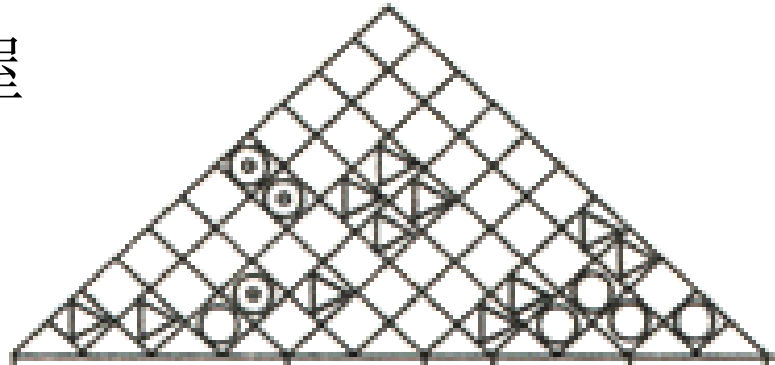



图 质量屋矩阵

# 图 蜡烛产品的质量屋



| | | | 蜡 | | | | 蜡芯 | | 芯+ | 染料 | | 香料 | | 顾客需求重要性 | 今天的我们 | 制造商1 | 制造商2 | 将来的我们 | 改进率 | 销售点 | 得分 | 得分% |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 粘度 | 熔点 | 直径 | 形状 | 蜡芯数目 | 处理 | 浓度 | 颜色 | 浓度 | 表面 | 挥发性 | | | | | | | | | |
| 销售功能 | 审美 | 视觉上有吸引力 | | | ▷ | ◎ | ▷ | ▷ | ▷ | ◎ | ◎ | ▷ | | 10 | 9 | 8 | 8 | 9 | 1.0 | 1.0 | 10 | 16 |
| | | 有香味 | | | | | | | | | | ◎ | ◎ | 4 | 4 | 6 | 8 | 8 | 2.0 | 1.2 | 9 | 15 |
| 使用功能 | 亮度 | 火焰大 | ○ | | ○ | | ◎ | ▷ | ▷ | | | | | 8 | 7 | 7 | 8 | 7 | 1.0 | 1.0 | 8 | 13 |
| | 方便性 | 无烟 | ▷ | | | | ◎ | ◎ | | | | | | 6.5 | 8 | 6 | 7 | 8 | 1.0 | 1.2 | 8 | 12 |
| | | 不滴蜡 | ○ | ○ | ◎ | | ◎ | | ◎ | | | | | 4.6 | 8 | 8 | 10 | 9 | 1.1 | 1.0 | 5 | 8 |
| | 效率 | 燃烧时间长 | ○ | ◎ | ◎ | ◎ | ◎ | ◎ | ◎ | | | | | 5 | 3 | 9 | 9 | 9 | 3.0 | 1.5 | 23 | 36 |
| | | 得分 | 183 | 348 | 451 | 468 | 637 | 461 | 425 | 144 | 48 | 151 | 135 | | | | | | | | Σ=63 | Σ=100 |
| | | 得分% | 5.8 | 10.1 | 13.1 | 13.6 | 18.5 | 13.4 | 12.5 | 4.2 | 1.4 | 4.4 | 3.9 | | | | | | | | | |
| | | 成本% | 43 | | | | 2 | | 15 | 9 | | 31 | | | | | | | | | | |
| | | 测量单位 | | F | cm | | 个 | | | | | | | | | | | | | | | |
| | | 目标值 | | 150 | | | 5 | | | | | | | | | | | | | | | |

# 5.3 获取需求

- QFD对需求的分类：
  - 常规需求：
    - 也称普通需求，包含客户对项目的最基本需求，是客户对整个项目最为关心的部分。
  - 期望需求：
    - 客户可能没有表达明确或没有明确提出的需求，但是会让客户提升对项目的满意度。
  - 意外需求：
    - 也称兴奋需求，如果实现会给客户带来惊喜，但是如果无法实现也不会受到客户责备。

# 5.3 获取需求

- QFD通过客户访谈和观察、调查以及检查历史数据为需求手机活动获取原始数据。然后将这些数据翻译成需求表——客户意见表，并由客户评审。
- 接下来使用各种图表、矩阵和评估方法抽取期望的需求并努力导出令人兴奋的需求。

# 5.3 获取需求

⑥ 用户场景

- 通常被称为用例，它提供了系统将如何被使用的描述
- 关注用户将如何使用该功能

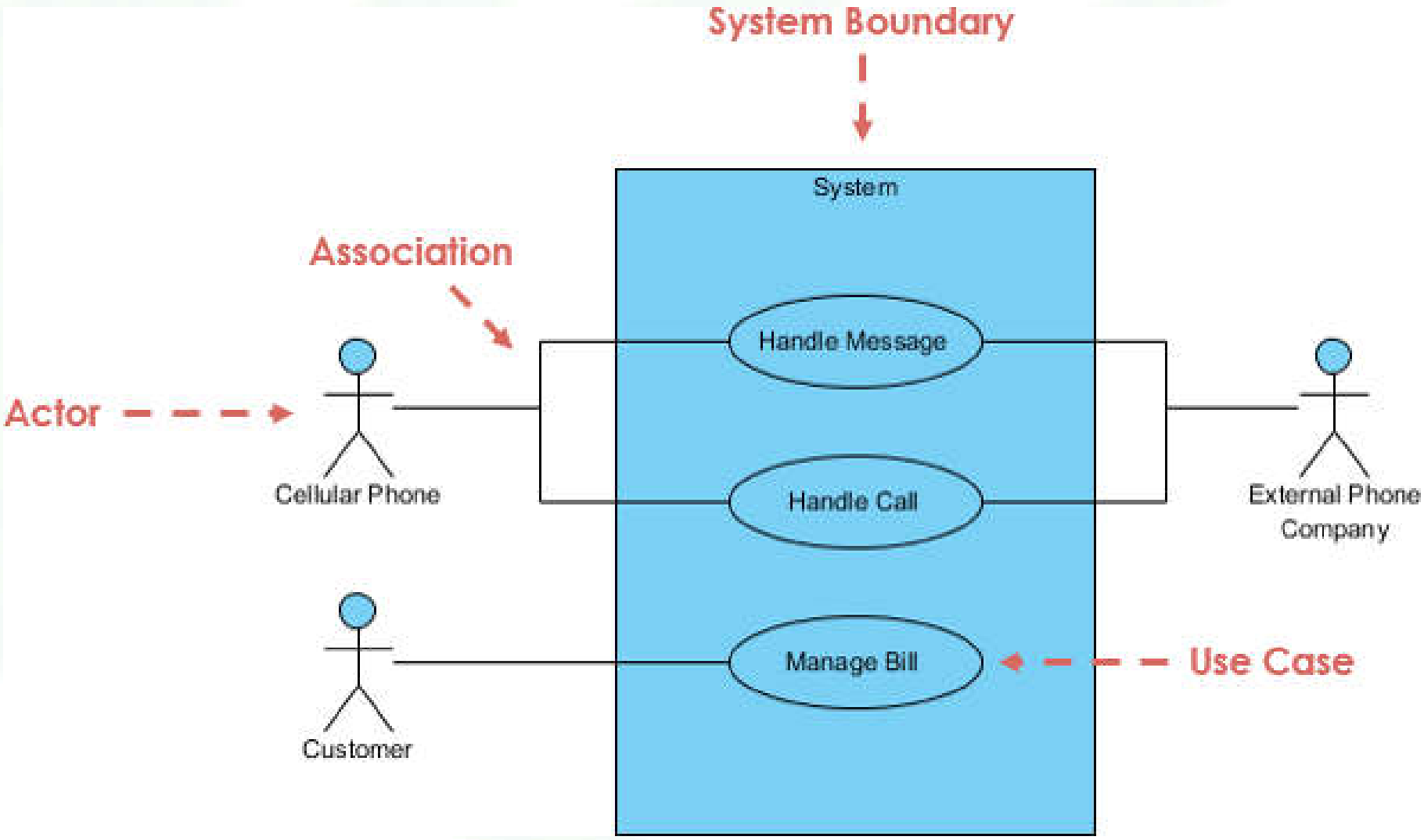
# 用户场景

- 导出工作产品
  - 对于大多数系统而言，工作产品包括：
    - 必要性和可行性陈述
    - 系统或产品范围的界限说明
    - 参与需求导出的客户、用户和其他共利益者的列表
    - 系统技术环境的说明
    - 需求列表以及每个需求适用的领域限制
    - 一系列使用场景，有助于深入了解系统或产品在不同运行环境下的使用
    - 任何能够更好地定义需求的原型

# 用户场景

- 用例讲述了能表达主体场景的故事：
  - 最终用户（扮演多种可能角色中的一个）如何在一特定环境下和系统交互。

# 用例图


System Boundary
System
Association
Actor
Handle Message
Handle Call
Manage Bill
Use Case
Cellular Phone
External Phone Company
Customer

| USE CASE 5 | Buy Goods | |
|---|---|---|
| **Description** | Buyer issues request directly to our company, expects goods shipped and to be billed. | |
| **Used by** | Manage customer relationship (use case 2) | |
| **Preconditions** | We know Buyer, their address, and needed buyer information. | |
| **Success End Condition** | Buyer has goods, we have money for the goods. | |
| **Failed End Condition** | We have not sent the goods, Buyer has not spent the money. | |
| **Actors** | Buyer, any agent (or computer) acting for the customer.<br>Credit card company, bank, shipping service | |
| **Trigger** | purchase request comes in. | |
| **DESCRIPTION** | **Step** | **Action** |
| | 1 | Buyer calls in with a purchase request |
| | 2 | Company captures buyer's name, address, requested goods, etc. |
| | 3 | Company gives buyer information on goods, prices, delivery dates, etc. |
| | 4 | Buyer signs for order. |
| | 5 | Company creates order, ships order to buyer. |
| | 6 | Company ships invoice to buyer. |
| | 7 | Buyers pays invoice. |
| **EXTENSIONS** | **Step** | **Branching Action** |
| | 3a | Company is out of one of the ordered items:<br>3a1. Renegotiate order. |
| | 4a | Buyer pays directly with credit card:<br>4a1. Take payment by credit card (use case 44) |
| | 7a | Buyer returns goods:<br>7a. Handle returned goods (use case 105) |
| **VARIATIONS** | | **Branching Action** |
| | 1 | Buyer may use<br>phone in,<br>fax in,<br>use web order form,<br>electronic interchange |
| | 7 | Buyer may pay by<br>cash or money order<br>check<br>credit card |

# 5.3 获取需求

## ⑥ 敏捷需求获取

在敏捷过程中，通过向所有利益者询问，生成用户故事以获取需求。

批评人士认为这种方法常缺少对全局商业目标和非功能需求的考量。在某些情况下，需要返工并考虑性能和安全问题。另外，用户故事可能无法为随时间发展的系统提供足够的基础。

# 敏捷需求获取

- 一个好的用户故事应该是 - **INVEST**：
  - **I ndependent 独立的**：应该是自包含的方式，允许在不依赖于彼此释放
  - **N egotiable 可协商的**：只捕捉用户需求的本质，为谈话留出空间。用户故事不应该像合同一样写。
  - **V aluable 有价值的**：为最终用户提供价值。
  - **E stimable 可评估的**：用户故事必须能够被估计，以便可以正确地确定优先级并适合短跑。
  - **S mall 小粒度的**：用户故事是一小部分工作，可以在大约3到4天内完成。
  - **T estable 可测试的**：必须通过预先编写的验收标准确认用户故事。

# 5.3 获取需求

## ⑥ 面向服务的方法

- 关注由应用系统所定义的服务。
- 从基于服务的模型中获取的需求细化了由应用场景所衍生出的服务。
- 每个触点都代表用户与系统的一次交互，从而获得所需的服务。

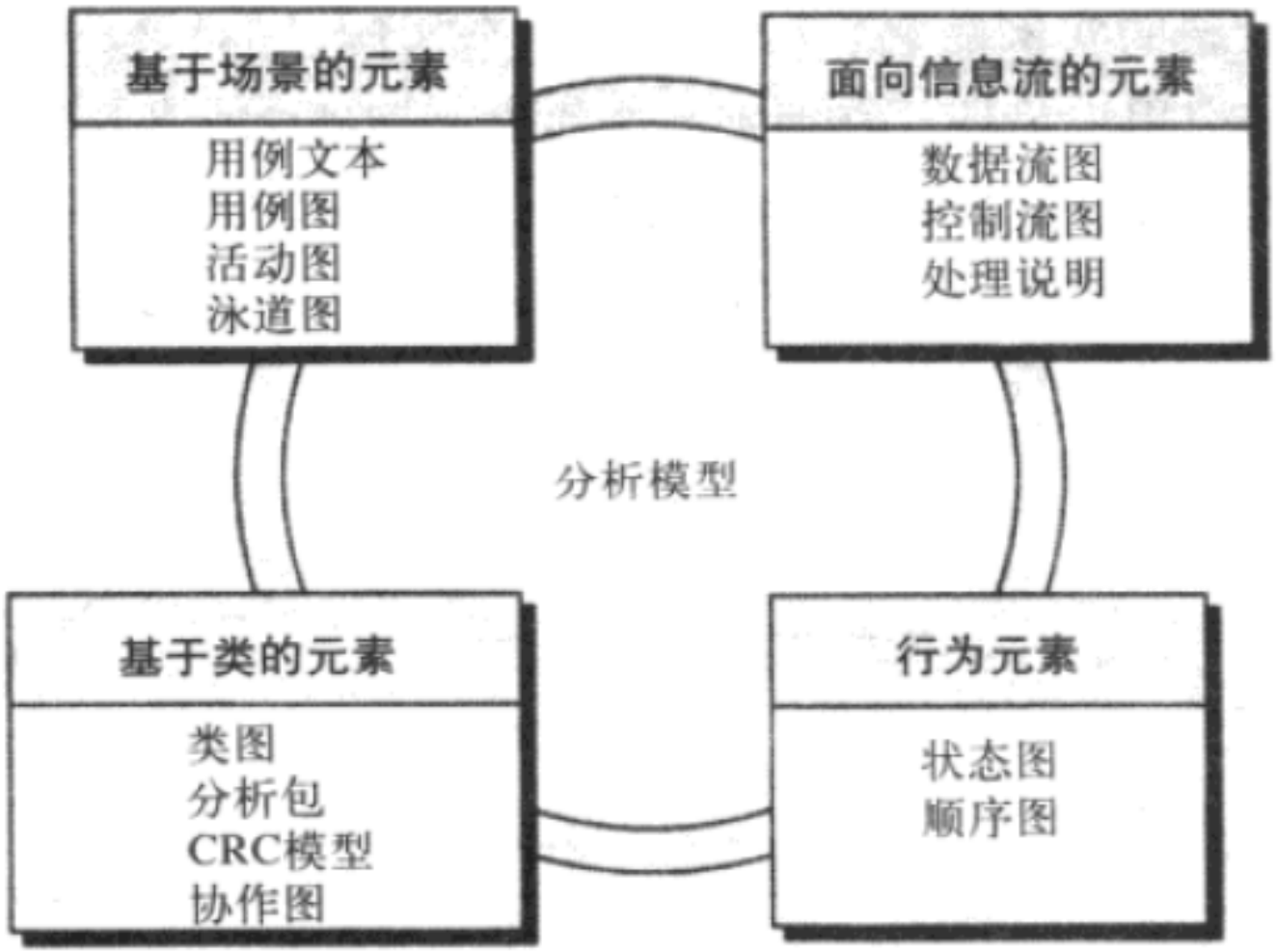
# 5.5 构建需求模型

- 分析模型的目的是为基于计算机的系统提供必要的信息、功能和行为域的说明。
- 分析模型是任意给定时刻的需求快照（更新）

# 需求模型的元素

- 表达模式选择一种或多种
- 需求模型的特定元素取决于将要使用的分析建模方法

# 需求模型的元素


基于场景的元素
用例文本
用例图
活动图
泳道图
面向信息流的元素
数据流图
控制流图
处理说明
分析模型
基于类的元素
类图
分析包
CRC模型
协作图
行为元素
状态图
顺序图

# 5.6 协商需求

- 协调过程的目的是保证所开发的项目计划，在满足利益相关者要求的同时反应软件团队所处真实世界的限制（如时间、人员、预算）
- 最好能利益相关者和软件团队“双赢”

# 5.7 确认需求

- 模型元素创建后，需要检查一致性、是否有遗漏以及歧义性。
  - 每项需求都和系统或产品的整体目标一致吗？
  - 所有的需求都已经在相应的抽象层上说明了吗？
  - 需求是真正必需的，还是另外加上去的，有可能不是系统目标所必需的特性吗？
  - 每项需求都有界定且无歧义吗？
  - 每项需求都有归属吗？--一个明确的人

# 5.7 确认需求

- 有需求与其他需求冲突吗？
- 在系统或产品所处的技术环境下每个需求都能够实现吗？
- 一旦实现后，没想需求是可测试的吗？
- 需求模型恰当地反映了将要构建系统的信息、功能和行为吗？
- 需求模型是否已经使用合适的方式“分割”，能够逐步地揭示详细的信息系统吗？
- 已经使用了需求模式简化需求模型吗？所有的模式都已经恰当地确认了吗？所有的模式都和客户的需求一致吗？

# 作业

- 1 （理解）为什么大量的软件开发人员没有足够重视需求工程？以前有没有什么情况让你可以跳过需求工程？
- 2 （理解）想出三个以上在需求起始阶段可以要问利益相关者的“与情景无关的问题”