

需求分析

需求分析的意义

软件需求的深入理解是软件开发工作获得成功的前提条件，不论我们把设计和编码做得如何出色，不能真正满足用户需求的程序只会令用户失望，给开发带来烦恼。

项目失败的原因分析

No.	Top 10 Factors		平均值
1	Inadequate requirements specification	不充分的需求规范	4.5
2	Changes in requirements	需求的改变	4.3
3	Shortage of systems engineers	缺乏系统工程师	4.2
4	Shortage of software managers	缺乏了解软件特性的经理人	4.1
5	Shortage of qualified project managers	缺乏合格的项目经理	4.1
6	Shortage of software engineers	缺乏软件工程师	3.9
7	Fixed - price contract	固定价合同	3.8
8	Inadequate communications for system integration	系统集成阶段,交流与沟通不充分	3.8
9	Insufficient experience as team	团队缺乏经验	3.6
10	Shortage of application domain experts	缺乏应用领域专家	3.6
Scale: 5 = Very Serious 3 = Serious 1 = No Serious			

Source: Carnegie-Mellon University(卡内基·梅隆), Software Engineering Institute

什么是软件需求

- IEEE软件工程标准词汇表 (1997) :
 - (1)用户解决问题或达到目标所需的条件或能力。
 - (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。

什么是软件需求

- 其他几种关于“需求”的定义：
 - 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。
 - 需求是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。

什么是软件需求

需求分析是**软件定义时期**的最后一个阶段，它的基本任务**不是确定系统怎样完成**它的工作，**而是确定系统必须完成**哪些工作，也就是对目标系统提出完整、准确、清晰、具体的要求。

并在在需求分析阶段结束之前，由系统分析员写出软件需求规格说明书，以书面形式准确地描述软件需求。即：

---- 准确地回答 “系统必须做什么?” 。

需求分析的几个问题

需求分析的复杂性

- 需求分析的复杂性和面临的困难
 - 片面, 不完全
 - 模糊, 不准确
 - 不一致, 歧义
 - 需求复杂和庞大
- 因此**必须使用系统的方法**、借助于一系列行之有效的技术和工具进行软件**需求分析**



客户是这样描述需求的



项目经理是这么理解的



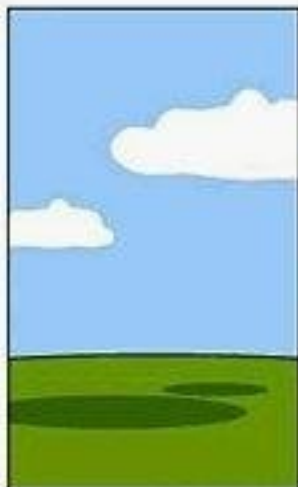
分析师是这么设计的



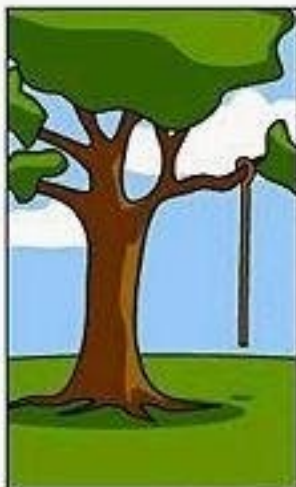
程序员是这么编写的



商业顾问是这么描绘的



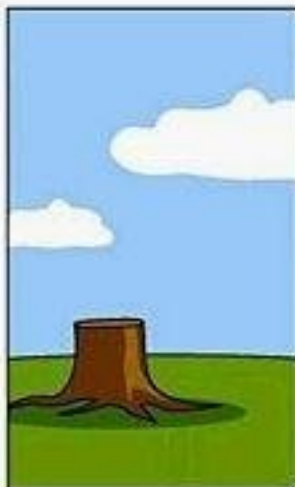
项目书写出来是这样的



操作中用了这样的工具



客户是这么建造的



提供的支持就是这个样子

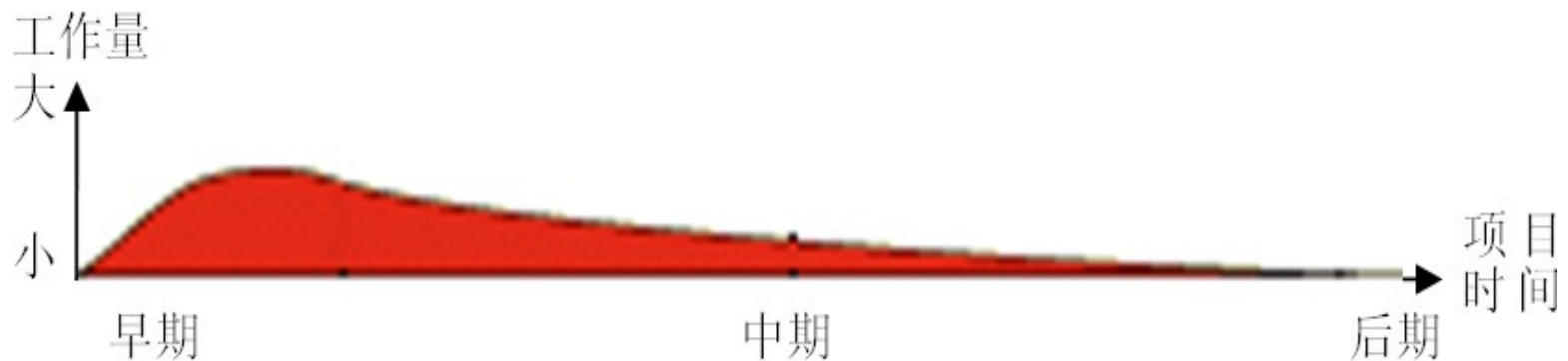
软件需求的描述

软件需求的描述

- 结构化语言、PDL
- 图形化表示
- 数学描述（形式化语言描述）

什么时候进行需求分析

- 项目的早期阶段？



- 贯穿于整个软件开发过程的需求活动

参与需求分析的人有哪些，场所在哪

- 参与需求分析的人

- 系统分析师、需求阐释者、客户代表、用户代表、开发方领导、项目经理、架构设计师、领域专家、财务人员、市场人员、软件质量保证人员、程序员、测试人员、部署人员、技术文档编写人员、培训人员等。

- 需求分析的场所

- 调研时，在客户现场
- 编纂软件需求规约文档时，可以在开发单位
- 复审相关的需求文档时，根据需要来安排

需求分析的目的与准则

- 目的

- 准确地回答 “**系统必须做什么?**” 这个问题。
- 对目标系统提出完整、准确、清晰、具体的要求。

需求分析的目的与准则

• 准则

- 必须理解并描述问题的信息域，根据这条准则应该建立数据模型。
- 必须定义软件应完成的功能，这条准则要求建立功能模型。
- 必须描述作为外部事件结果的软件行为，这条准则要求建立行为模型。
- 必须对描述信息、功能和行为的模型进行分解，用层次的方式展示细节。

需求分析

3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体-联系图

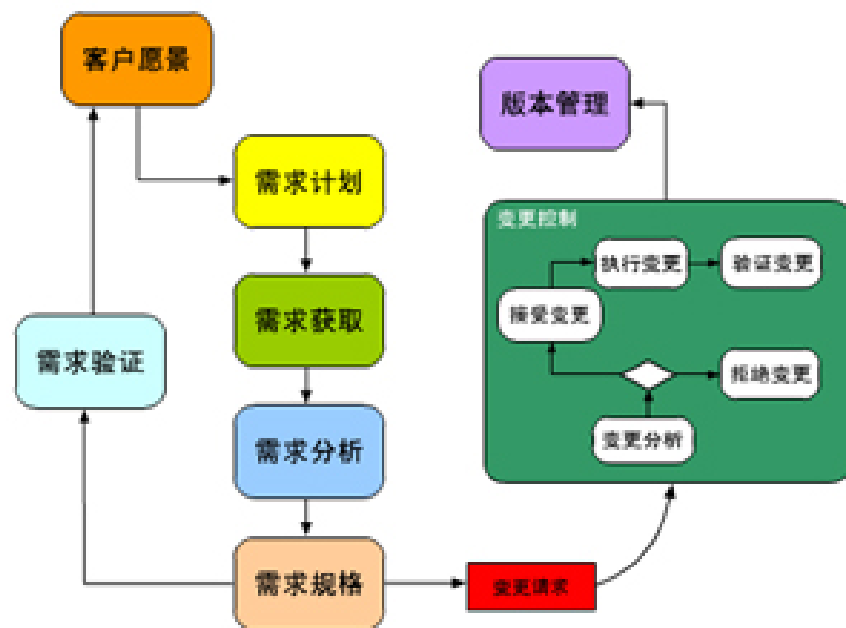
3.5 数据规范化

3.6 状态转换图+有穷状态机

3.7 其他图形工具

3.8 验证软件需求

3.9 小结



需求分析

3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体-联系图

3.5 数据规范化

3.6 状态转换图+有穷状态机

3.7 其他图形工具

3.8 验证软件需求

3.9 小结

需求分析

3.1 需求分析的任务

确定对系统的综合要求
分析系统的数据要求
导出系统的逻辑模型
修正系统开发计划

需求分析

3.1 需求分析的任务

确定对系统的综合要求

分析系统的数据要求

导出系统的逻辑模型

修正系统开发计划

确定对系统的综合要求

- **功能需求**：划分出系统必须完成的所有功能。
- **性能需求**：指定系统必须满足的定时约束或容量约束，通常包括速度(响应时间)、信息量速率、主存容量、磁盘容量、安全性等方面的需求。
- **可靠性和可用性需求**：定量地指定系统的可靠性，它量化了用户可以使用系统的程度。
- **出错处理需求**：说明系统对环境错误应该怎样响应，这类错误并不是由该应用系统本身造成的。
- **接口需求**：接口需求描述应用系统与它的环境通信的格式。
- **约束**：描述在设计或实现应用系统时应遵守的限制条件。常见的约束有：**精度；工具和语言约束；设计约束；应该使用的标准；应该使用的硬件平台。**
- **逆向需求**：**说明软件系统不应该做什么。**
- **将来可能提出的要求**：应该明确地列出那些虽然不属于当前系统开发范畴，但是据分析将来很可能会提出来的要求。

需求分析

3.1 需求分析的任务

确定对系统的综合要求

分析系统的数据要求

导出系统的逻辑模型

修正系统开发计划

分析系统的数据要求

- 采用建立数据模型的方法
 - 数据字典：可以全面准确地定义数据，但不够形象直观。
 - 层次方框图和Warnier图：描绘数据结构，有助于提高可理解性。
 - 数据结构规范化：能够减少数据冗余，避免出现插入异常或删除异常以及简化修改数据的过程。

需求分析

3.1 需求分析的任务

确定对系统的综合要求

分析系统的数据要求

导出系统的逻辑模型

修正系统开发计划

导出系统的逻辑模型

- 综合上述两项分析的结果可以导出系统的详细的逻辑模型，描述这个逻辑模型通常用：
 - 数据流图
 - 实体-联系图
 - 状态转换图
 - 数据字典
 - 处理算法

需求分析

3.1 需求分析的任务

确定对系统的综合要求

分析系统的数据要求

导出系统的逻辑模型

修正系统开发计划

修正系统开发计划

- 根据在分析过程中获得的对系统的更深入更具体的了解，可以比较准确地估计系统的成本和进度，修正以前制定的开发计划。

软件产品的利益相关者

1. 用户：或称最终用户，直接使用软件系统的人。
2. 顾客：购买这个软件或者根据合同或规定接收软件的人。
3. 市场分析者：代表“典型用户”的需求。
4. 监管机构
5. 系统/应用集成商
6. 软件团队
7. 软件工程师

软件产品的利益相关者

你要写一个中学生学习英语的软件，找谁做用户调研？

软件产品的利益相关者

你要写一个中学生学习英语的软件，找谁做用户调研？

- 中学生 - 最终用户。
- 家长 - 掏钱的人。
- 学校老师 - 有影响力的人。

有哪些层次的软件需求

1. 业务需求 (business requirement)
2. 用户需求 (user requirement)
3. 功能需求 (functional requirement)
4. 非功能需求 (non-functional requirement)
 - 过程需求
 - 产品需求
 - 外部需求

有哪些层次的软件需求

1. **业务需求**：反映了组织机构或客户对系统或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。
2. **用户需求**：描述了用户使用产品必须要完成的任务，可在用例模型或方案脚本中予以说明。
3. **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。

有哪些层次的软件需求

案例：一个小型超市需要一个商品的查询系统。

业务需求：

进货人员需要查询商品库存以便保证及时进货；
收款员需要查询商品的销售价格以便结账；
经理需要查询商品的销售及盈利情况。

用户需求：

这三类用户怎样去查询系统，查询哪些信息，还需要做哪些操作。

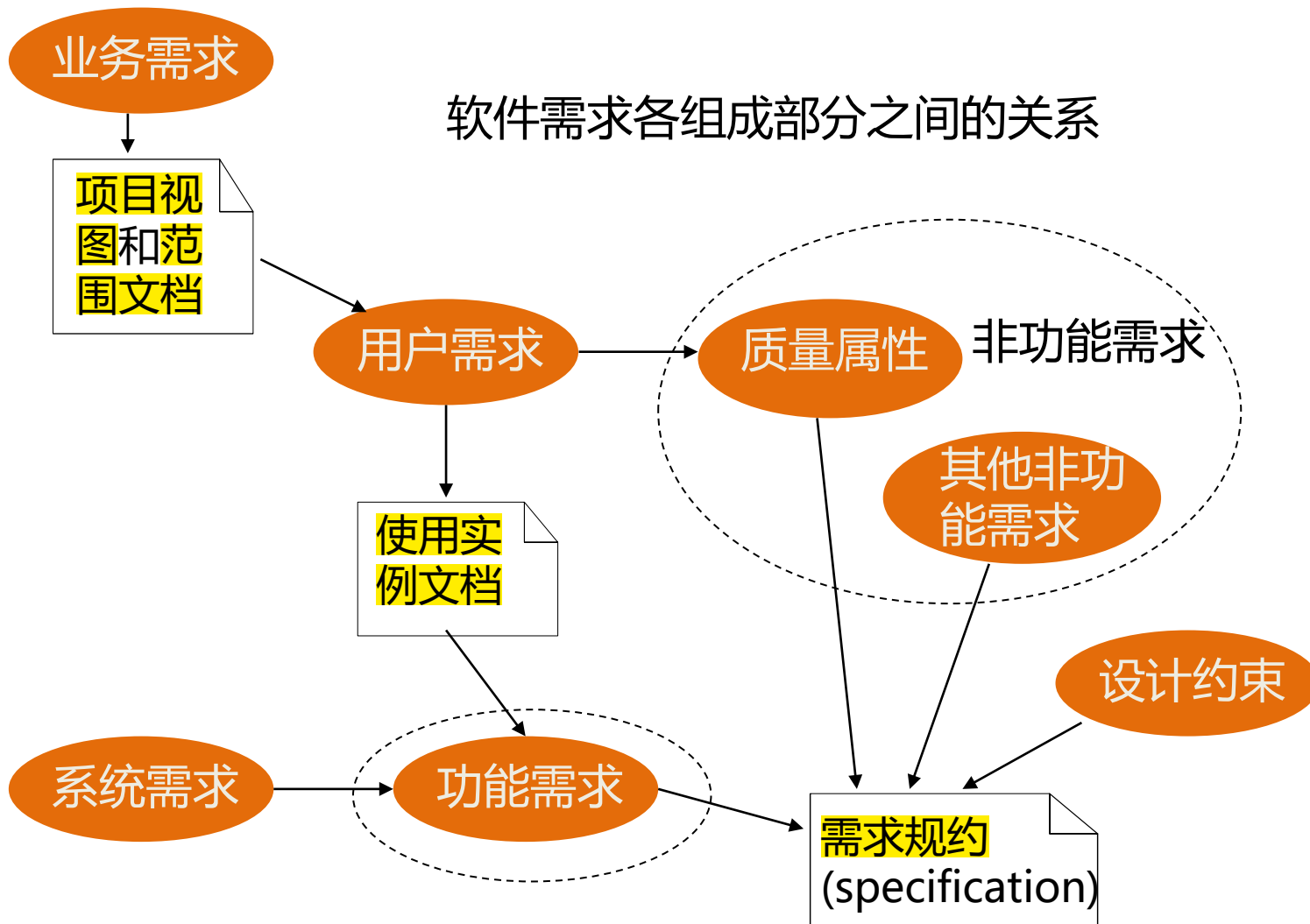
有哪些层次的软件需求

4. **非功能需求**：是从各个角度对系统的约束和限制，反映了应用对软件系统质量和特性的额外要求。

- **过程需求**：有交付、实现方法和标准等需求；
- **产品需求**：包含性能、可用性、实用性、可靠性、可移植性、安全保密性、容错性等方面的需求；
- **外部需求**：有法规、成本、操作性等需求。

5. **综合需求**

有哪些层次的软件需求



需求分析

3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体-联系图

3.5 数据规范化

3.6 状态转换图+有穷状态机

3.7 其他图形工具

3.8 验证软件需求

3.9 小结

软件开发的过程，就是“用户最需要的东西”在下面这一链条中传送、转换、实现、扭曲或丢失的过程。

用户最需要的

- >>用户表达出来的

- >> 软件团队能理解的+团队的商业目标

- >>软件团队成员具体表达出来的(规格说明书)

- >>在各种约束条件下，具体执行表达出来的(代码)

- >>验证通过的(测试)

- >>通过各种渠道告诉目标用户(发布/推广)

- >>用户终于能用上了，但是他们不满意



客户是这样描述需求的



项目经理是这么理解的



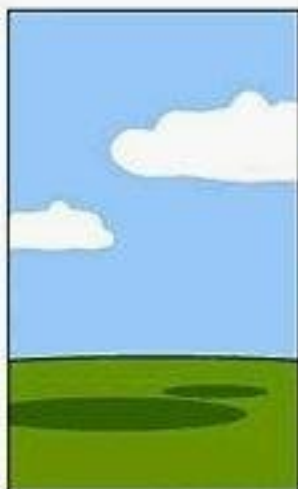
分析师是这么设计的



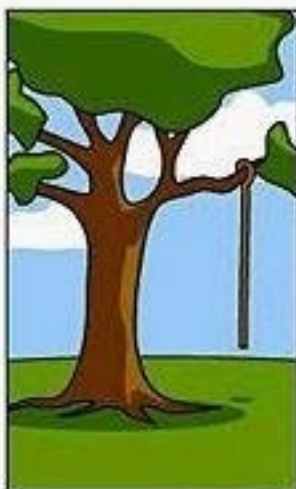
程序员是这么编写的



商业顾问是这么描绘的



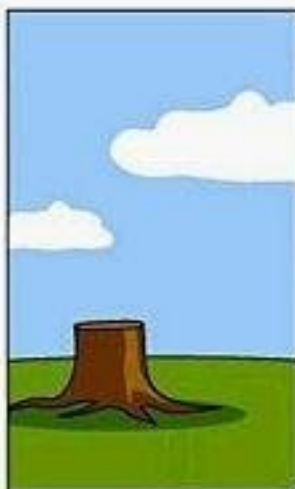
项目书写出来是这样的



操作中用了这样的工具



客户是这么建造的



提供的支持就是这个样子

你以为这样就结束了吗？



客户是这样描述需求的



项目经理是这么理解的



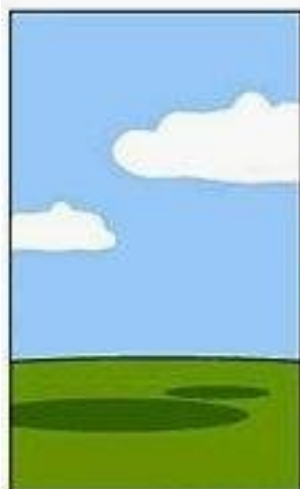
分析师是这么设计的



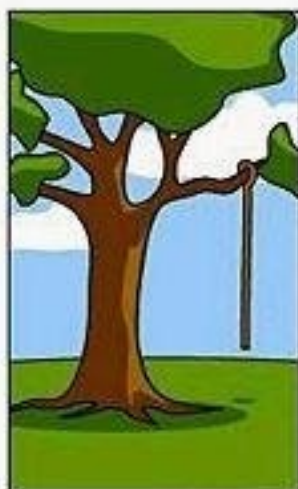
程序员是这么编写的



商业顾问是这么描绘的



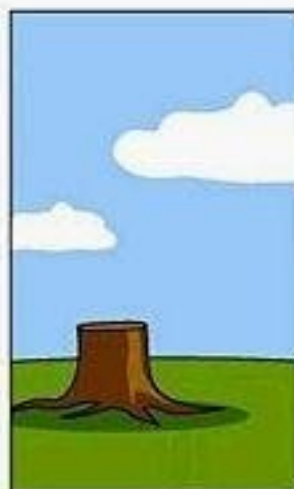
项目书写出来是这样的



操作中用了这样的工具



客户是这么建造的



提供的支持就是这个样子



而这才是
客户真正需要的

需求分析的步骤

1. 获取和引导需求(Elicitation)
 - 来自利益相关者
 - 来自管理机构
 - 来自软件企业本身
 - 来自技术团队本身
2. 分析和定义需求(Analysis & Specification)
3. 验证需求(Validation)
4. 在产品的生命周期中管理需求(Management)

需求分析

3.2 与用户沟通获取需求的方法

访谈

面向数据自顶向下求精

简易的应用规格说明技术

快速建立软件原型

其他方法

需求分析

3.2 与用户沟通获取需求的方法

访谈

面向数据自顶向下求精

简易的应用规格说明技术

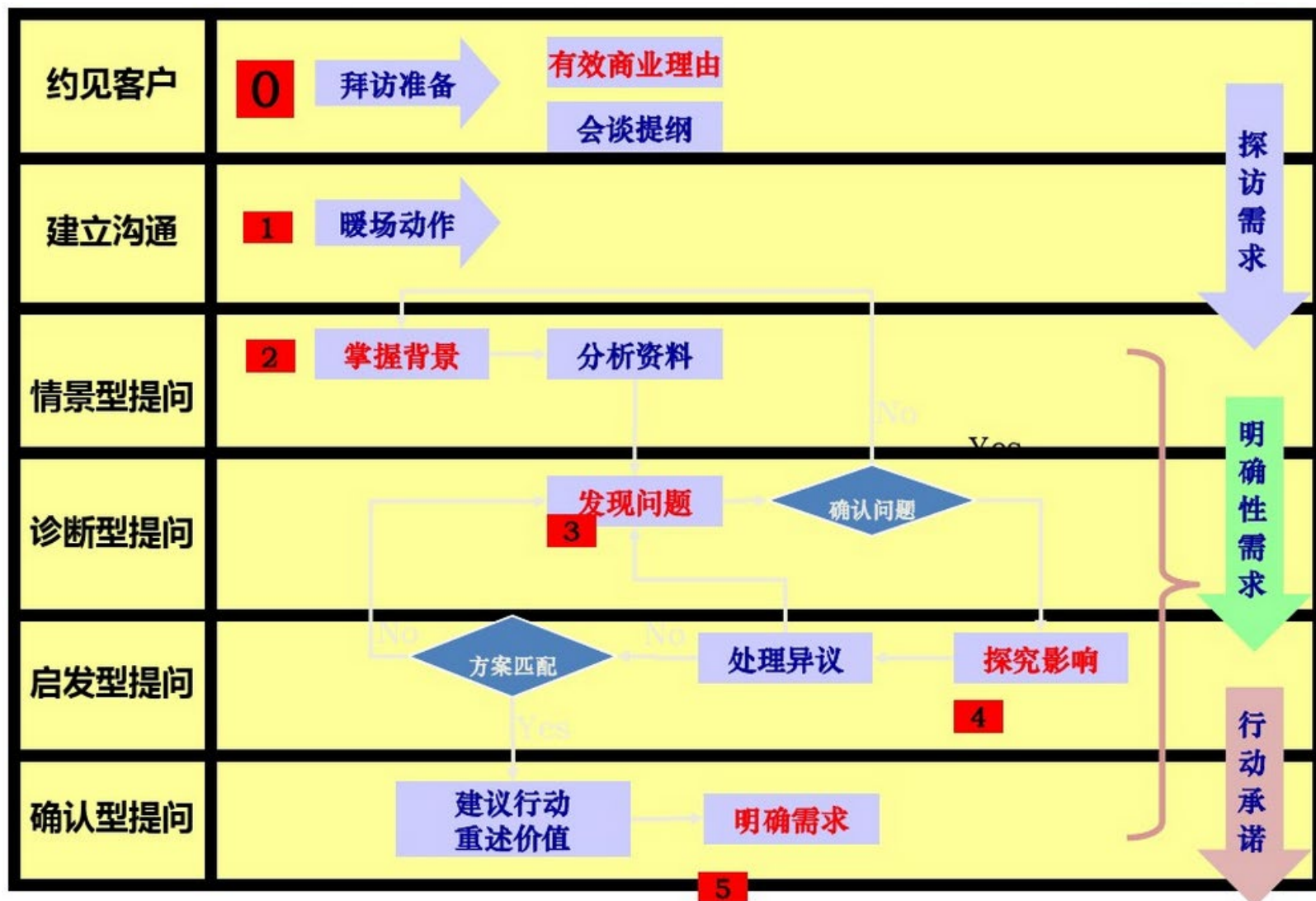
快速建立软件原型

其他方法

访谈

- **正式的：**系统分析员将提出一些事先准备好的具体问题。
- **非正式的：**分析员将提出一些用户可以自由回答的开放性问题，以鼓励被访问人员说出自己的想法。
- **情景分析：**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。
 - (1) 它能在某种程度上演示目标系统的行为，从而便于用户理解，而且还可能进一步揭示出一些分析员目前还不知道的需求。
 - (2) 由于情景分析较易为用户所理解，使用这种技术能保证用户在需求分析过程中始终扮演一个积极主动的角色。

访谈技巧



3.2 与用户沟通获取需求的方法——访谈

1. 焦点小组
2. 深入面谈
3. 卡片分类
4. 用户调查问卷
5. 用户日志研究
6. 人类学调查

3.2 与用户沟通获取需求的方法

1. 焦点小组
2. 深入面谈
3. 卡片分类
4. 用户调查问卷
5. 用户日志研究
6. 人类学调查



3.2 与用户沟通获取需求的方法——访谈

1. 焦点小组
2. 深入面谈
3. 卡片分类
4. 用户调查问卷
5. 用户日志研究
6. 人类学调查



3.2 与用户沟通获取需求的方法——访谈

1. 焦点小组
2. 深入面谈
3. 卡片分类
4. 用户调查问卷
5. 用户日志研究
6. 人类学调查



讨论→明晰定义→归类→排序

3.2 与用户沟通获取需求的方法——访谈

1. 焦点小组
2. 深入面谈
3. 卡片分类
4. 用户调查问卷
5. 用户日志研究
6. 人类学调查



3.2 与用户沟通获取需求的方法——访谈

1. 焦点小组

2. 深入面谈

3. 卡片分类

4. 用户调查问卷

5. 用户日志研究

6. 人类学调查

问卷调查一些常见的错误：

- a. 问题定义不准确
- b. 使用含糊不清的词
- c. 让用户花额外的努力来回答问题
- d. 问题带有引导性的倾向
- e. 问题涉及用户隐私、或者商业机密等

3.2 与用户沟通获取需求的方法——访谈

1. 焦点小组

2. 深入面谈

3. 卡片分类

4. 用户调查问卷

5. 用户日志研究

6. 人类学调查

问卷调查一些常见的方式：

a.全开放式问题

b.二项选择题

c.多项选择题

d.顺位选择题

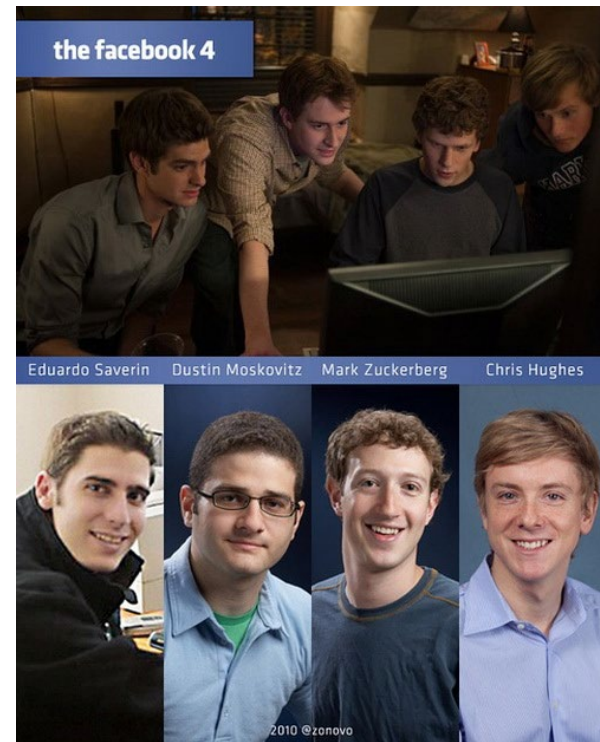
3.2 与用户沟通获取需求的方法——访谈

1. 焦点小组
2. 深入面谈
3. 卡片分类
4. 用户调查问卷
5. 用户日志研究
6. 人类学调查



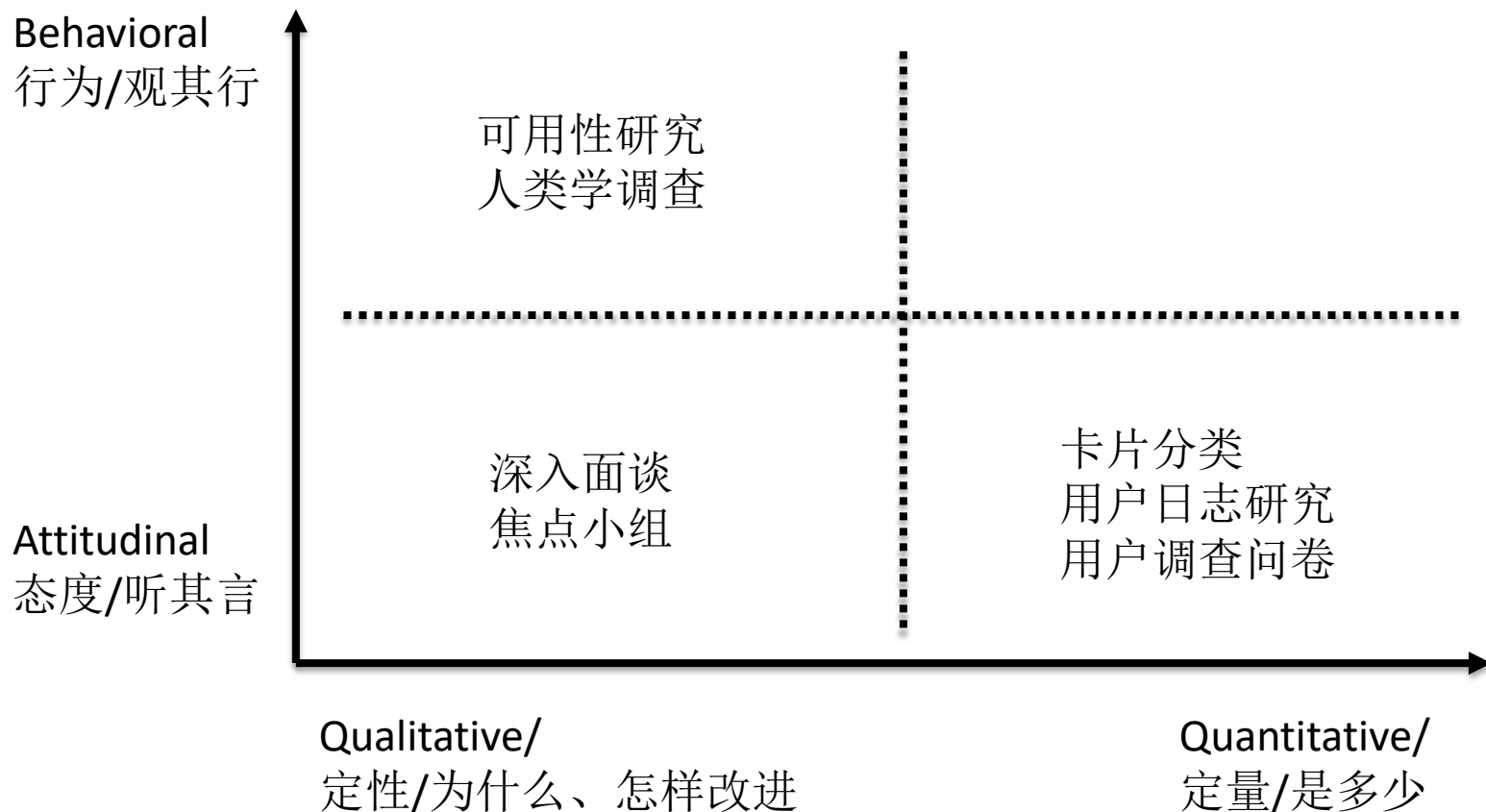
3.2 与用户沟通获取需求的方法——访谈

1. 焦点小组
2. 深入面谈
3. 卡片分类
4. 用户调查问卷
5. 用户日志研究
6. 人类学调查



《The Social Network》

3.2 与用户沟通获取需求的方法——访谈



做过头了会怎样？

互联网给我们带来了用户和数据，根据数据做分析和改进，Google是其中翘楚。

“Google公司的某团队在两种蓝色之间犹豫不，于是他们测试了两种蓝色之间的41种浓淡程度。最近我还和别人争论过一个线条究竟应该是3、4或5像素宽，然后还要证明我的论点。我在这种环境中已经不能正常工作。我已经厌倦了争执这些微不足道的设计问题。”

—— 谷歌首席设计师
Douglas Bowman



需求分析

3.2 与用户沟通获取需求的方法

访谈

面向数据自顶向下求精

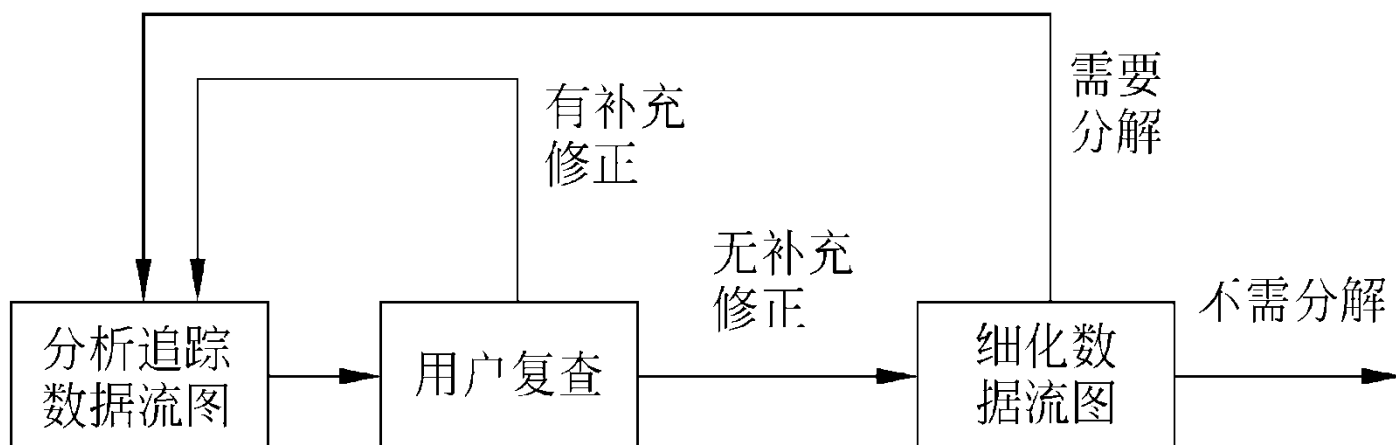
简易的应用规格说明技术

快速建立软件原型

其他方法

3.2 与用户沟通获取需求的方法——面向数据自顶向下求精

- **结构化分析方法**：通常从数据流图的输出端着手分析，沿数据流图从输出端往输入端回溯，确定每个数据元素的来源，与此同时也就初步定义了有关的算法。
 - 通常把分析过程中得到的有关数据元素的信息记录在数据字典中，把对算法的简明描述记录在IPO图中。



需求分析

3.2 与用户沟通获取需求的方法

访谈

面向数据自顶向下求精

简易的应用规格说明技术

快速建立软件原型

其他方法

3.2 与用户沟通获取需求的方法——简易的应用规格说明技术

- 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。
 - 初步的访谈，通过用户对基本问题的回答，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”。
 - 选定会议的时间和地点，并选举一个负责主持会议的协调人，邀请开发者和用户双方组织的代表出席会议，并在开会前几天预先把写好的产品需求分发给每位与会者认真审查。
 - 会议开始后，首先讨论是否需要这个新产品，如果大家都同意，每位与会者就应该把他们在会前准备好的列表展示出来供大家讨论。

3.2 与用户沟通获取需求的方法——简易的应用规格说明技术

- 大家共同创建一张组合列表。
 - 把与会者分成更小的小组，每个小组的工作目标是为每张列表中的项目制定小型规格说明并向全体与会者展示。
 - 每个与会者都制定出产品的一整套确认标准，并把自己制定的标准提交会议讨论，以创建出意见一致的确认标准。
 - 最后，由一名或多名与会者根据会议成果起草完整的软件需求规格说明书。
- **优点：**开发者与用户不分彼此，齐心协力，密切合作；即时讨论并求精；有能导出规格说明的具体步骤。

需求分析

3.2 与用户沟通获取需求的方法

访谈

面向数据自顶向下求精

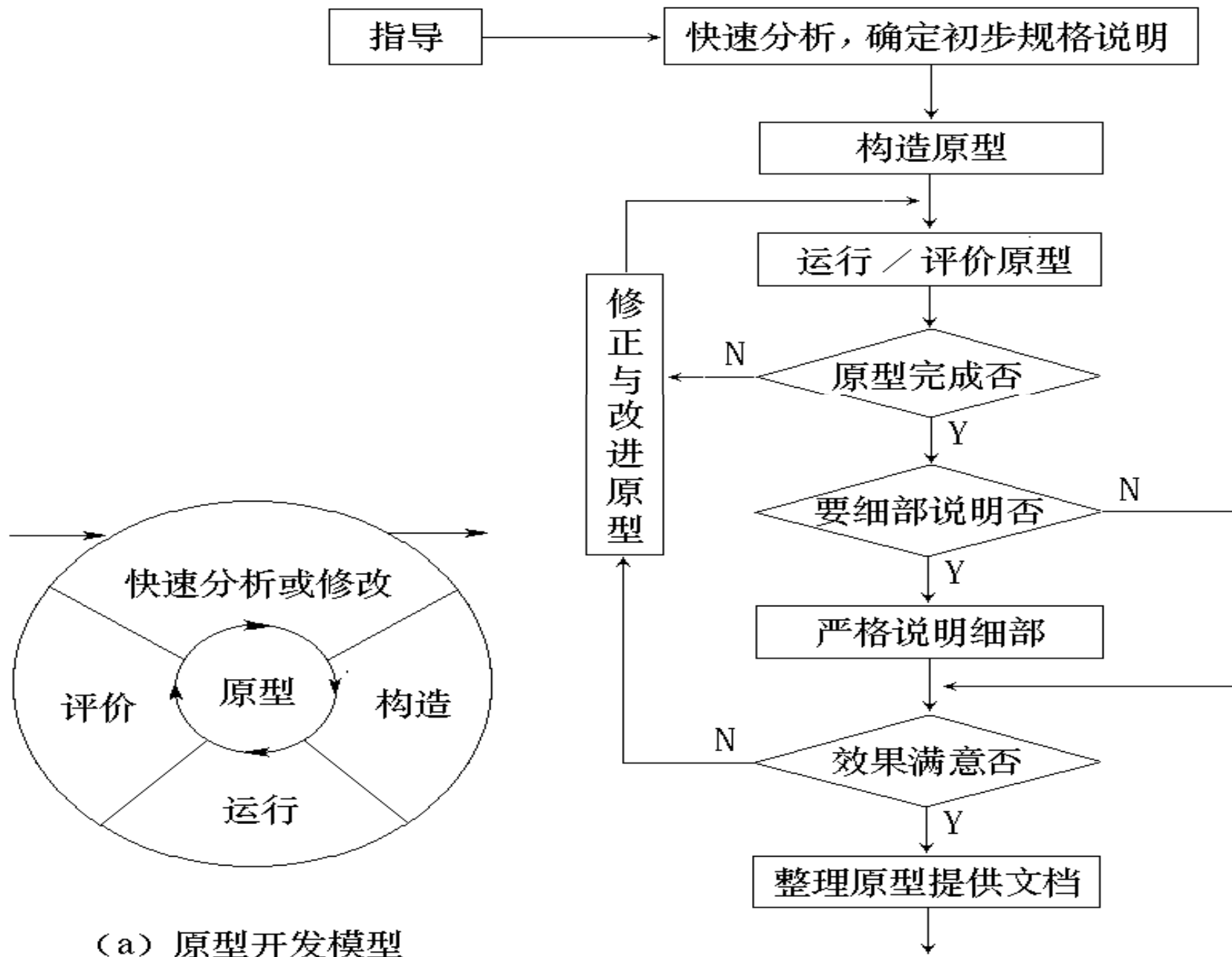
简易的应用规格说明技术

快速建立软件原型

其他方法

3.2 与用户沟通获取需求的方法——快速建立软件原型

- 快速建立起来的旨在演示目标系统主要功能的可运行的程序。
- 第一个特性是“快速”。
- 第二个特性是“容易修改”（原型的“修改—试用—反馈”过程可能重复多遍）。
- 使用的方法和工具
 - 第四代技术：包括众多数据库查询和报表语言、程序和应用系统生成器以及其他非常高级的非过程语言。
 - 可重用的软件构件：可以是数据结构(或数据库)，或软件体系结构构件(即程序)，或过程构件(即模块)。
 - 形式化规格说明和原型环境



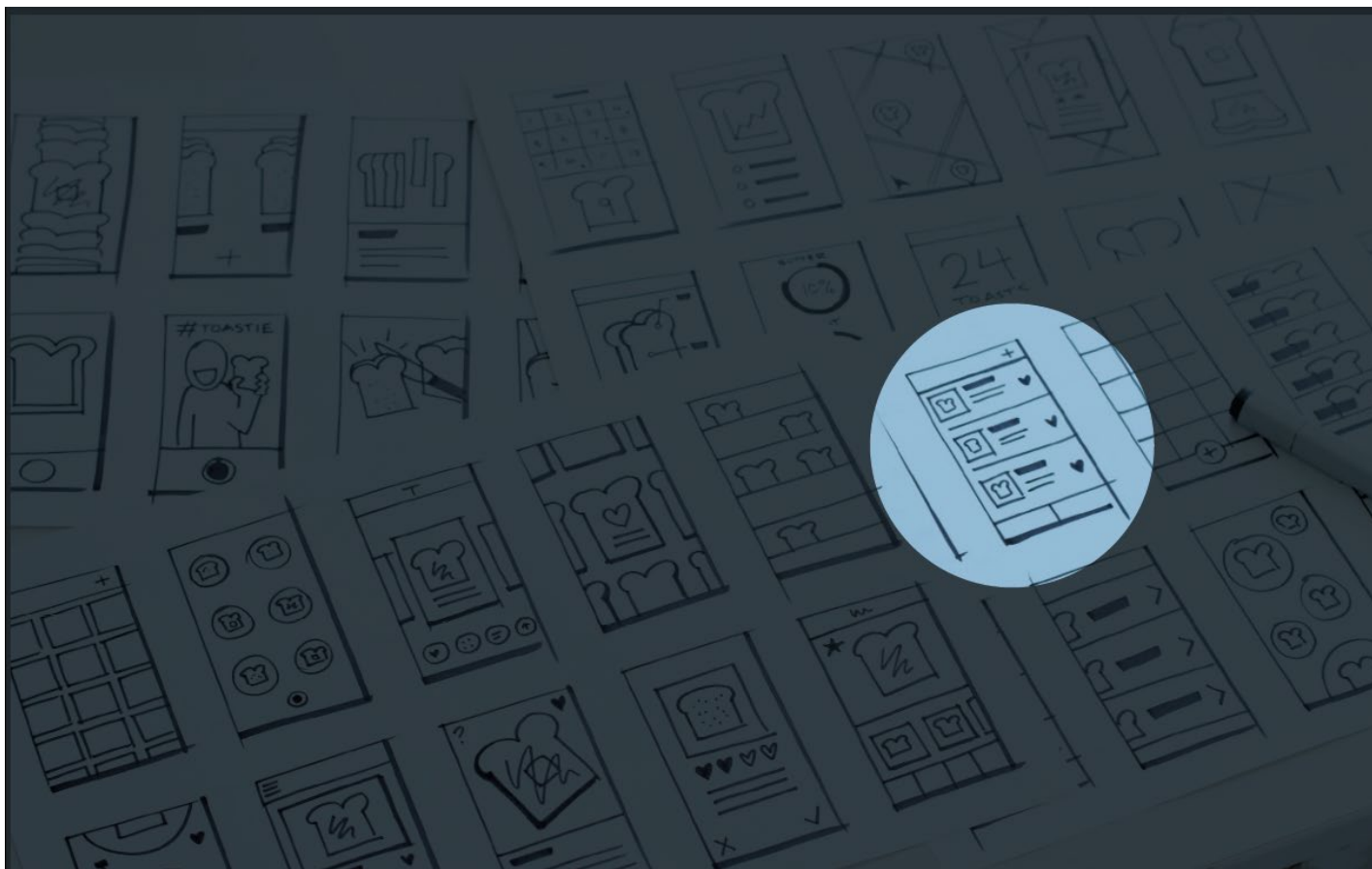
(a) 原型开发模型

(b) 模型的细化过程

3.2 与用户沟通获取需求的方法——快速建立软件原型



3.2 与用户沟通获取需求的方法——快速建立软件原型



需求分析

3.2 与用户沟通获取需求的方法

访谈

面向数据自顶向下求精

简易的应用规格说明技术

快速建立软件原型

其他方法

3.2 与用户沟通获取需求的方法——其他方法

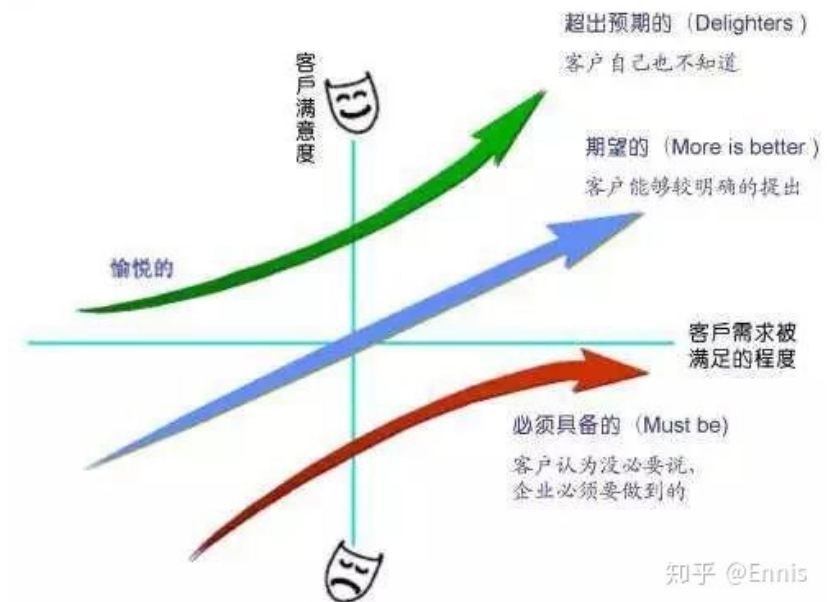
1. 竞争性需求分析的框架(NABCD)
2. 功能的定位和优先级（四象限方法）
3. 计划和估计 (Wideband Delphi)
4. 分而治之（任务划分技术WBS）

3.2 与用户沟通获取需求的方法——其他方法

1. 竞争性需求分析的框架(NABCD)
2. 功能的定位和优先级（四象限方法）
3. 计划和估计 (Wideband Delphi)
4. 分而治之（任务划分技术WBS）

3.2 与用户沟通获取需求的方法——NABCD

1. N (Need, 需求)
2. A (Approach, 做法)
3. B (Benefit, 好处)
4. C (Competitors, 竞争)
5. D (Delivery, 推广)



3.2 与用户沟通获取需求的方法——NABCD

1. N (Need, 需求)

① 假设用户的需求已经被满足；

2. A (Approach, 做法)

② 找到 “不消费的用户”

3. B (Benefit, 好处)

4. C (Competitors, 竞争)

5. D (Delivery, 推广)

3.2 与用户沟通获取需求的方法——NABCD

1. N (Need, 需求)

2. A (Approach, 做法)

3. B (Benefit, 好处)

4. C (Competitors, 竞争)

5. D (Delivery, 推广)

独特的办法:

- 技术上的
- 商业模式上的
- 地域的
- 人脉的
- 行业的
- 成本上的
-

3.2 与用户沟通获取需求的方法——NABCD

1. N (Need, 需求)

① 带给客户的好处

2. A (Approach, 做法)

② 迁移成本问题

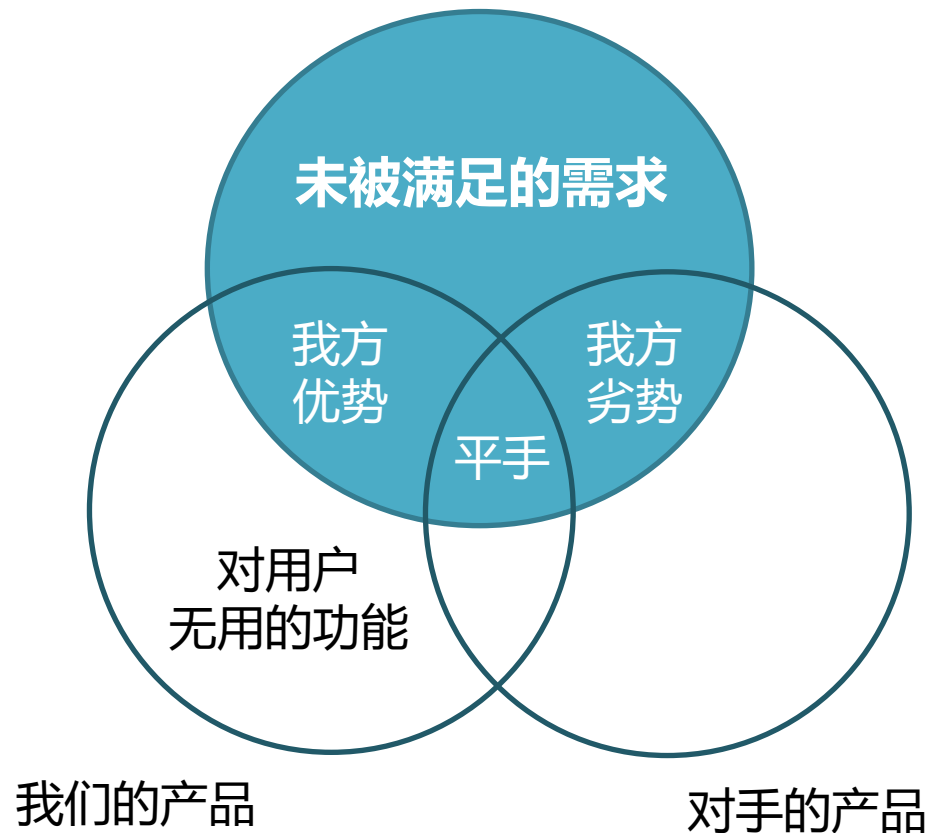
3. B (Benefit, 好处)

4. C (Competitors, 竞争)

5. D (Delivery, 推广)

3.2 与用户沟通获取需求的方法——NABCD

1. N (Need, 需求)
2. A (Approach, 做法)
3. B (Benefit, 好处)
4. C (Competitors, 竞争)
5. D (Delivery, 推广)



3.2 与用户沟通获取需求的方法——NABCD

1. N (Need, 需求)
2. A (Approach, 做法)
3. B (Benefit, 好处)
4. C (Competitors, 竞争)
5. D (Delivery, 推广)

- ① 假设用户的需求已经被满足;
- ② 找到 “不消费的用户”

3.2 与用户沟通获取需求的方法——NABCD

电梯演说

1. N（需求）
2. A（做法）
3. B（好处）
4. C（竞争）
5. D（推广）

我们的产品〈名称〉是为了解决〈目标用户〉的痛点，他们需要〈Need〉，但是现有的产品并没有很好地解决这些需求。我们有独特的办法〈Approach〉，它能给用户带来好处〈Benefit〉，远远超过竞争对手〈Competitor〉。同时，我们有高效率的推广〈Delivery〉方法，能很快地让目标户知道我们的产品，并进一步传播。

3.2 与用户沟通获取需求的方法——NABCD

电梯演说

1. N（需求）

2. A（做法）

3. B（好处）

4. C（竞争）

5. D（推广）

我们的〈功能改进〉是为了解决〈目标用户〉的痛点，他们**需要**〈Need〉，但是现有的方案并没有很好地解决这些需求。我们有**独特的办法**〈Approach〉，它能给用户带来**好处**〈Benefit〉，远远超过**竞争对手**〈Competitor〉，包括我们以前的版本。我们有**数据**〈Data〉（用户调查）支持我们的观点。我们相信它能给我们带来〈Data〉的业绩改善（用户量、使用时间、评价、收入等）。

3.2 与用户沟通获取需求的方法——其他方法

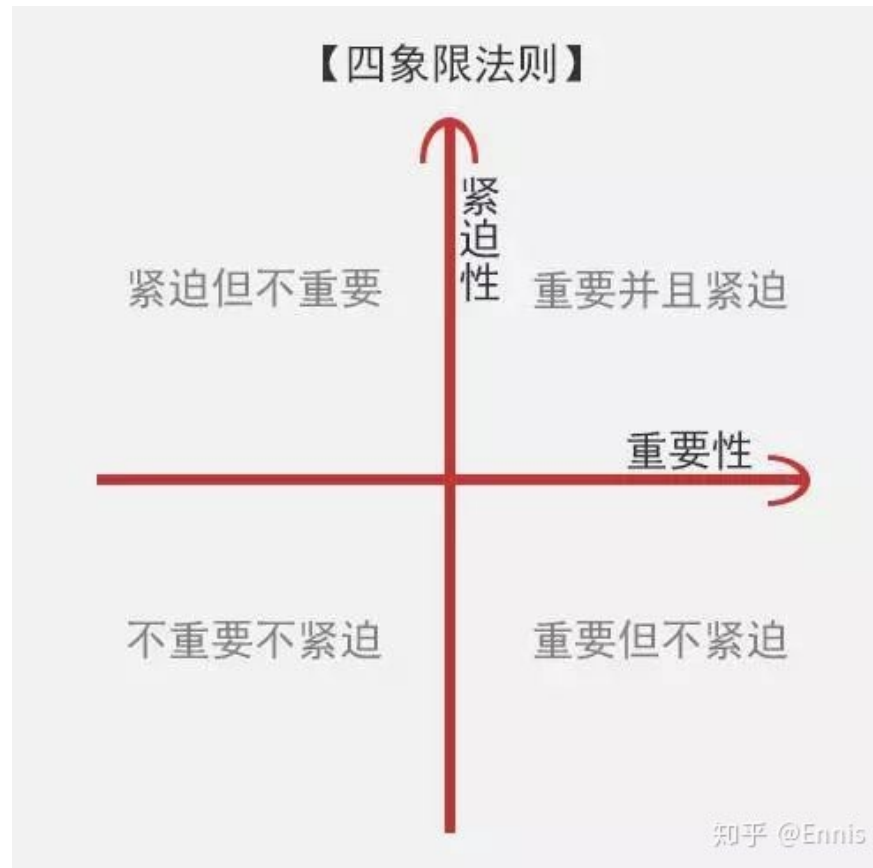
1. 竞争性需求分析的框架(NABCD)
2. 功能的定位和优先级（四象限方法、KANNO图）
3. 计划和估计 (Wideband Delphi)
4. 分而治之（任务划分技术WBS）

3.2 与用户沟通获取需求的方法——四象限

问：为什么你要做个功能(产品)，而不是别的功能(产品)

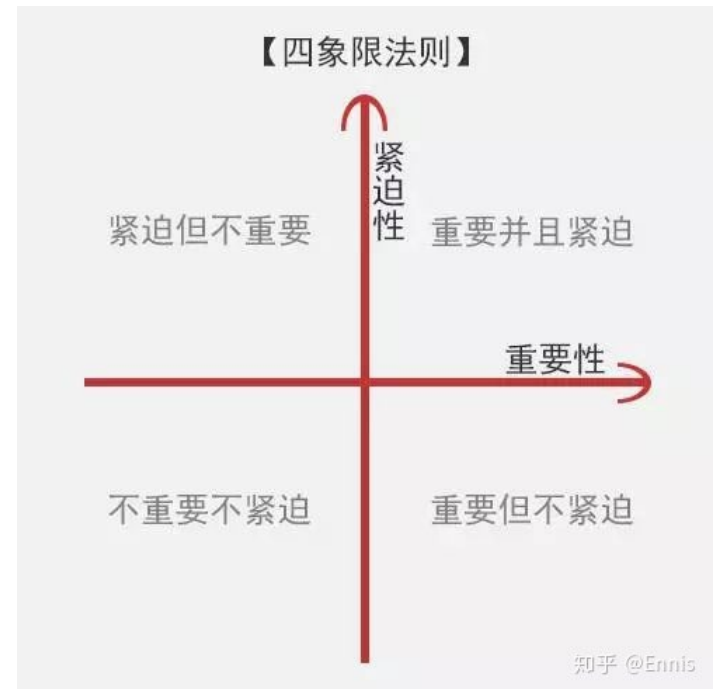
- 老板说啥就做啥
- 我来的时候，大家就做这个功能了，所以我也做
- 我觉得做这个功能很Happy，所以我做
- 别的产品通过这个功能赚钱，我们也做
- 别的产品显示不出我的技术水平，我不想做
- 这个做的人太多，我不做
- 这个做的人太少，我不做
-

3.2 与用户沟通获取需求的方法——四象限



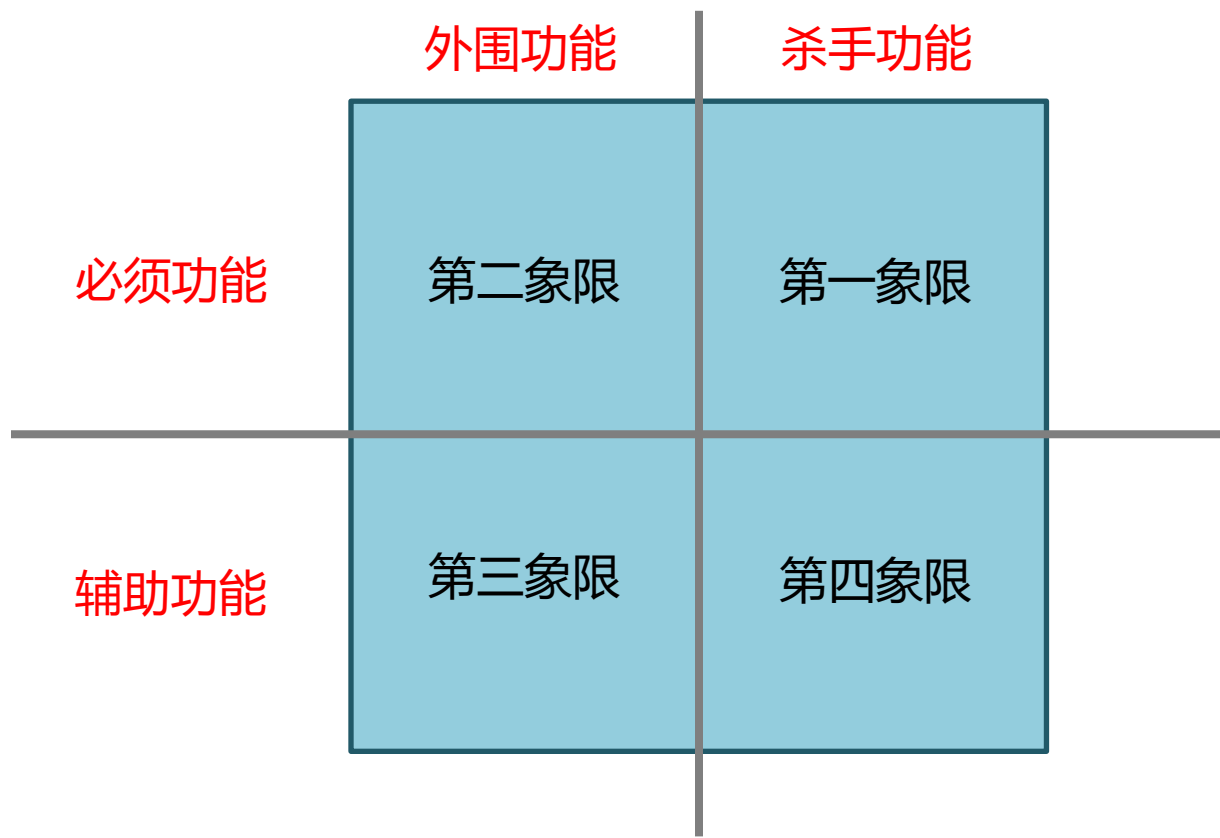
3.2 与用户沟通获取需求的方法——四象限

- 杀手功能 (Core)
- 外围功能 (Context)
- 必要需求 (Mission Critical)
- 辅助需求 (Enabling)



✓ 这四种划分结合起来，就得到了功能分析的四个象限。

3.2 与用户沟通获取需求的方法——四象限



3.2 与用户沟通获取需求的方法——四象限

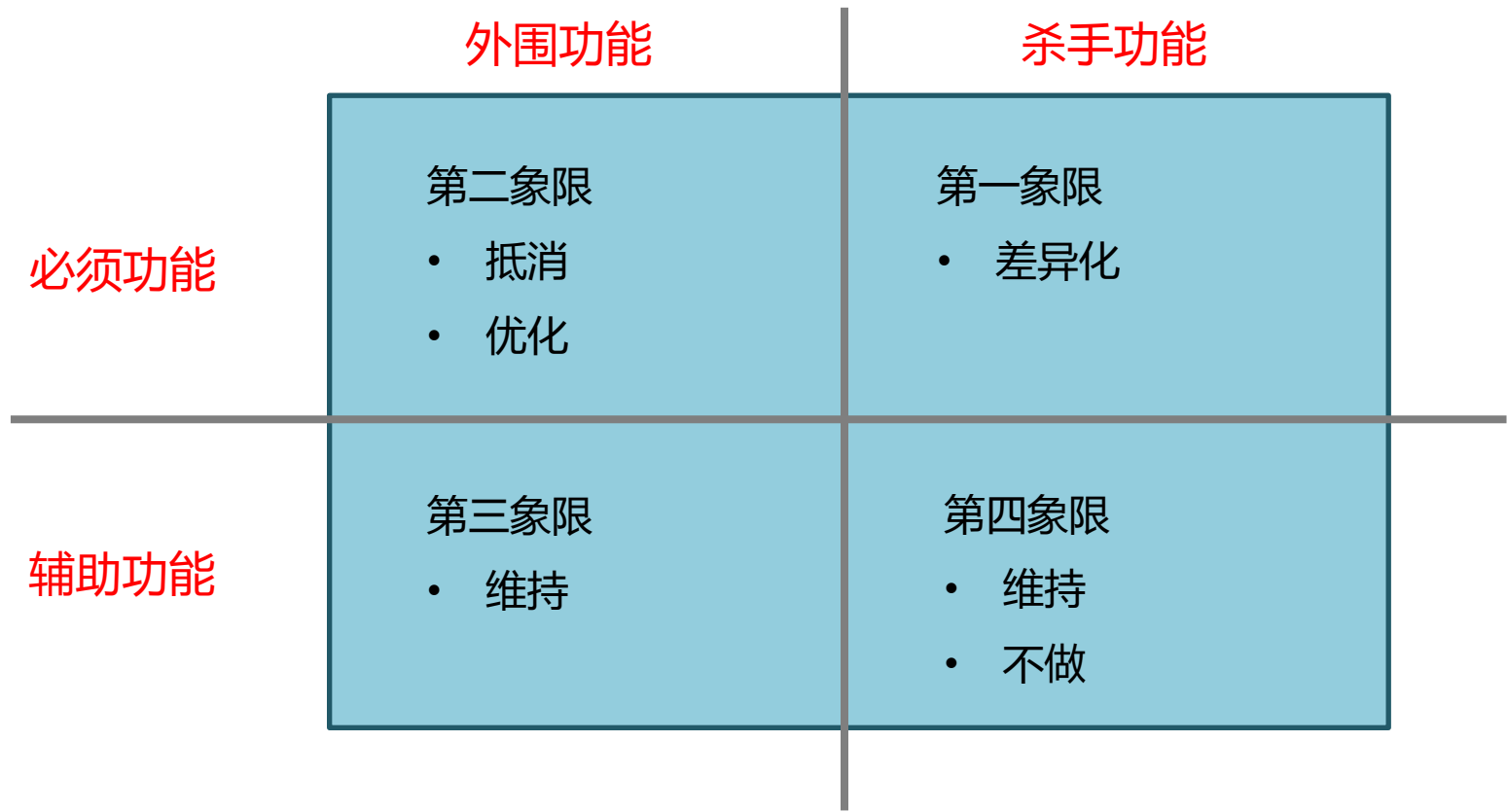


3.2 与用户沟通获取需求的方法——四象限

资源有限，我们对不同功能有哪些办法呢？

- **维持**——以最低成本维持此功能
- **抵消**——快速地达到“足够好”、“和竞争对手差不多”
- **优化**——花大力气做到并保持行业最好
- **差异化**——产生同类产品比不了的功能或优势(我有人无的优势，或者一个数量级以上的优势)
- **不做**——砍掉一个功能不定要所有的

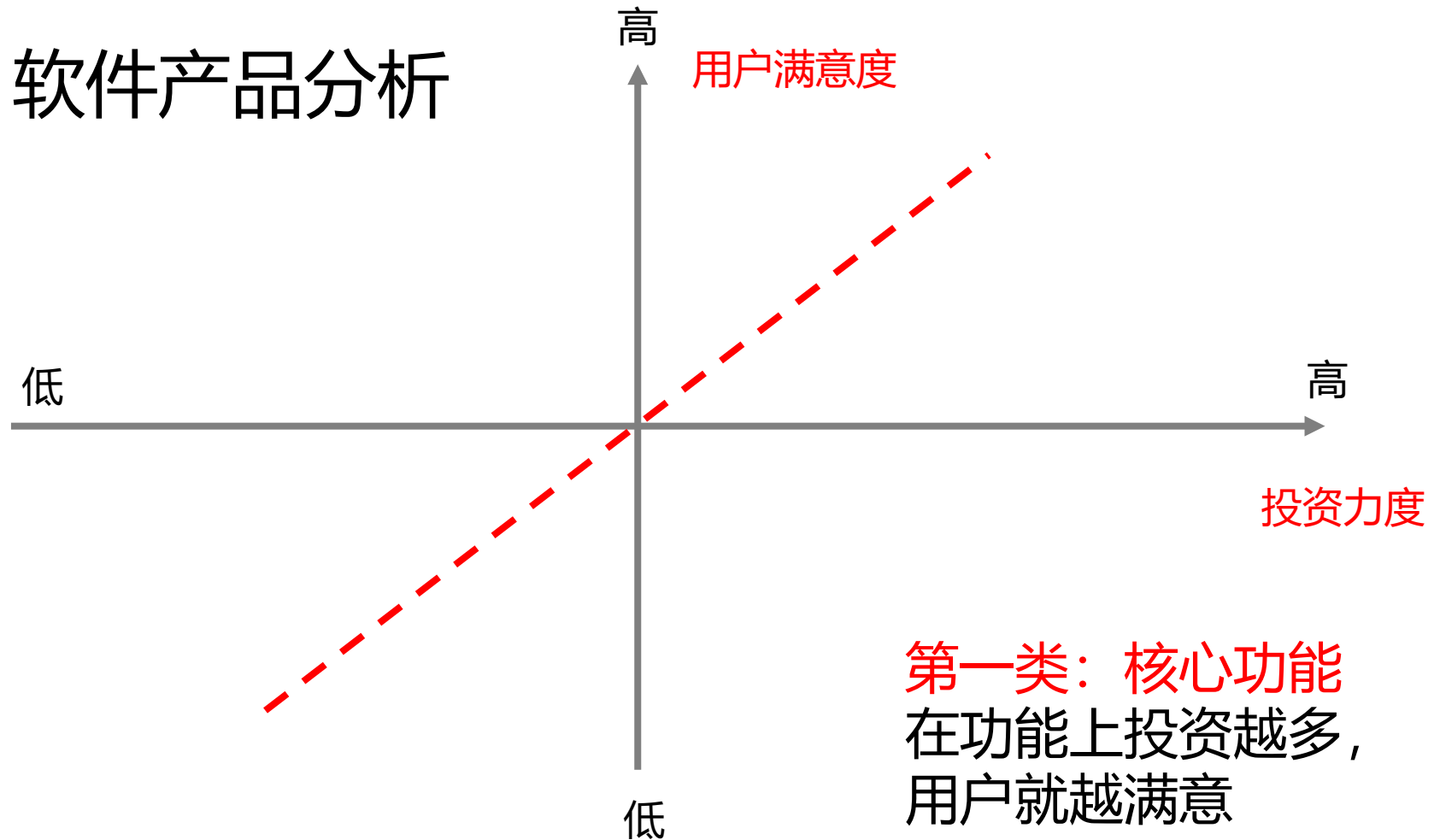
3.2 与用户沟通获取需求的方法——四象限



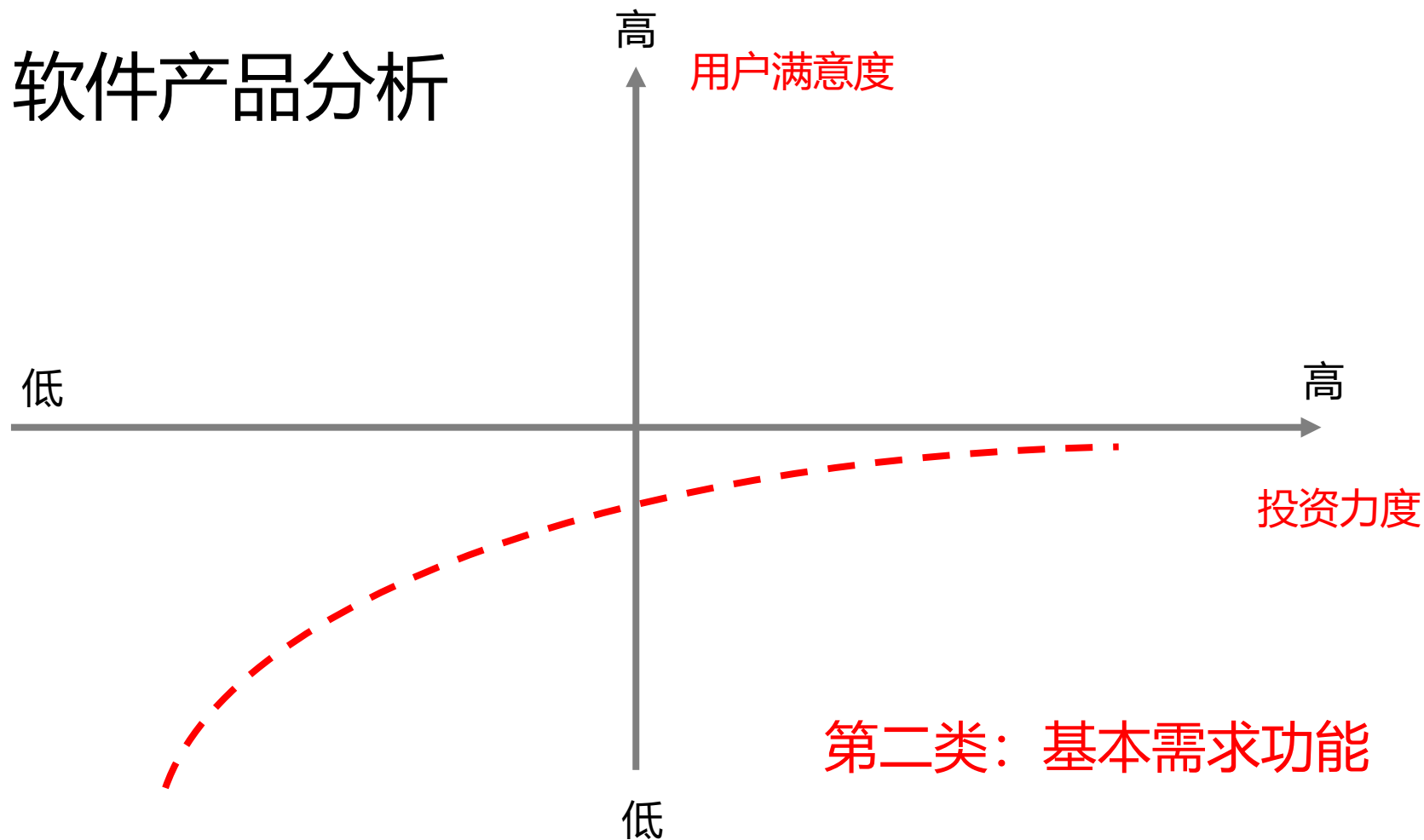
3.2 与用户沟通获取需求的方法——四象限



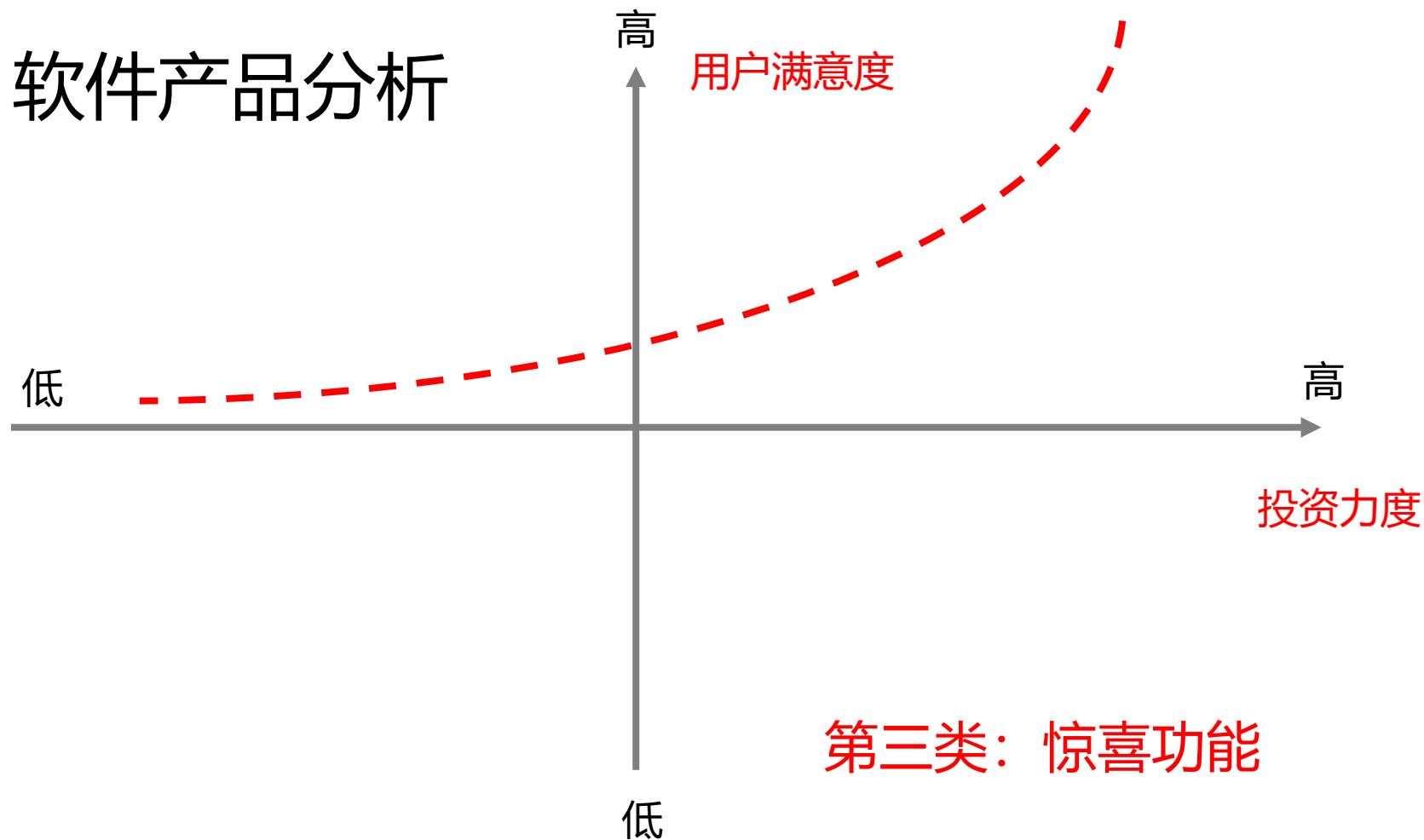
3.2 与用户沟通获取需求的方法——四象限



3.2 与用户沟通获取需求的方法——四象限

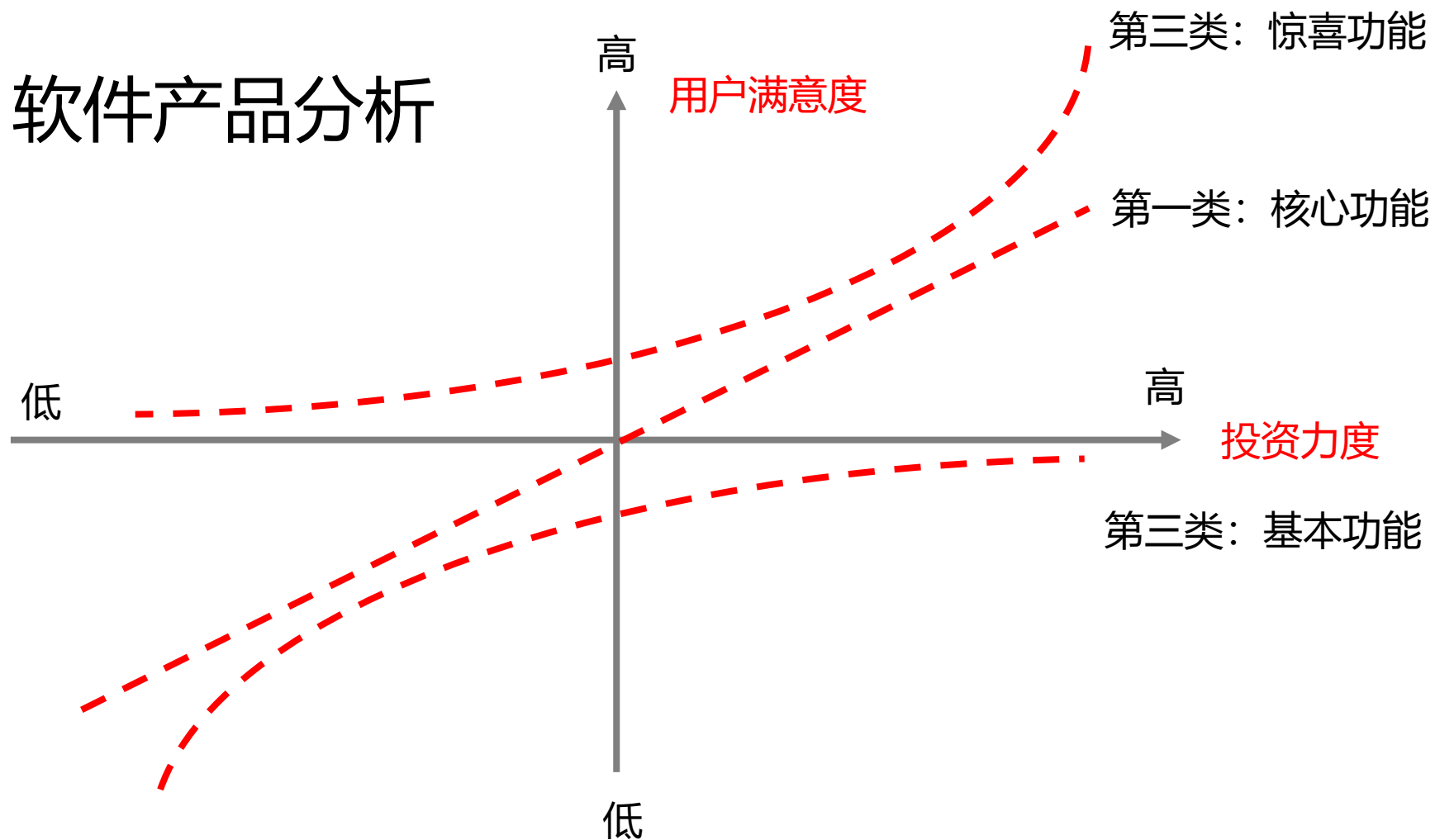


3.2 与用户沟通获取需求的方法——四象限

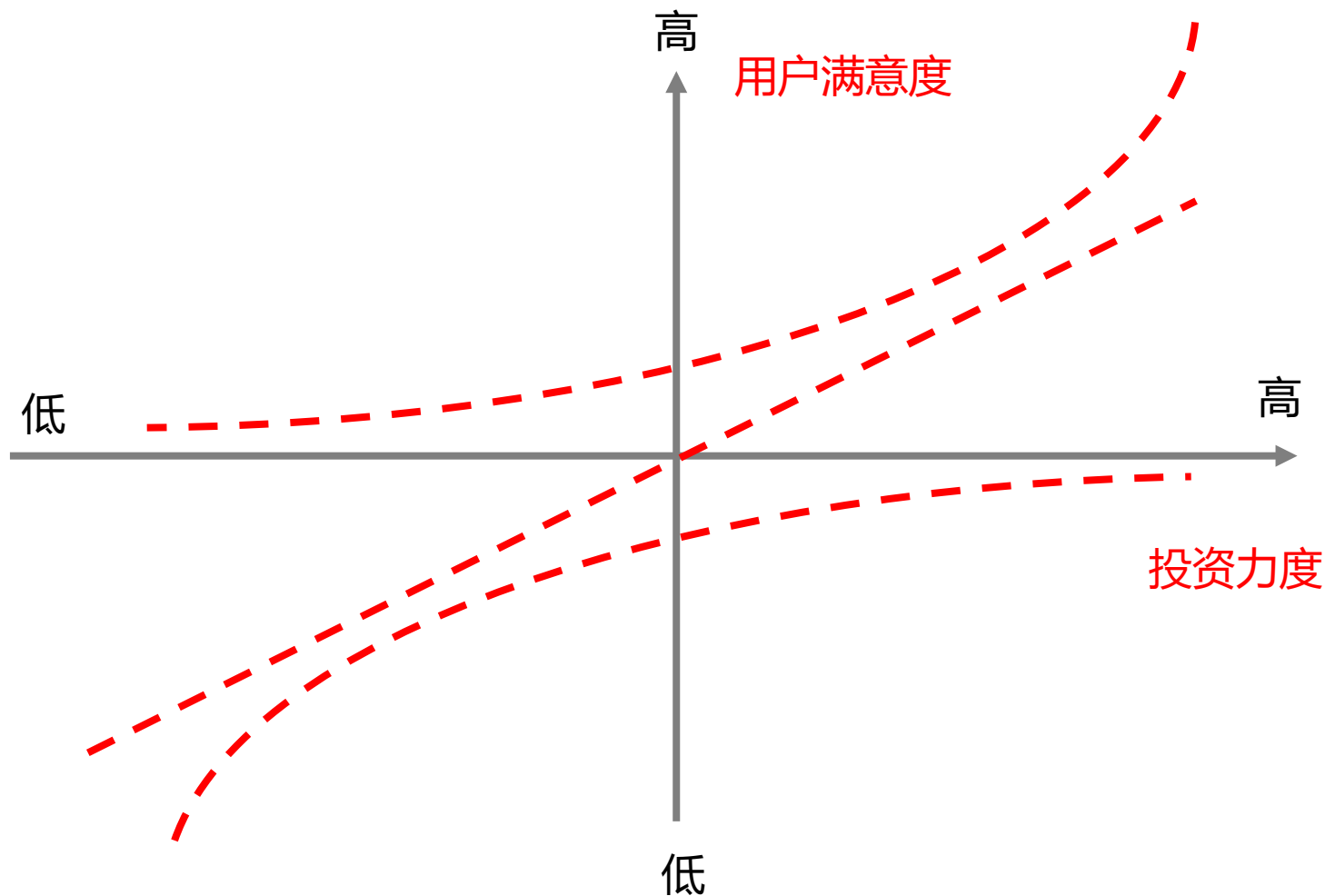


3.2 与用户沟通获取需求的方法——四象限

软件产品分析



3.2 与用户沟通获取需求的方法——四象限



3.2 与用户沟通获取需求的方法——其他方法

1. 竞争性需求分析的框架(NABCD)
2. 功能的定位和优先级（四象限方法）
3. 计划和估计 (Wideband Delphi)
4. 分而治之（任务划分技术WBS）

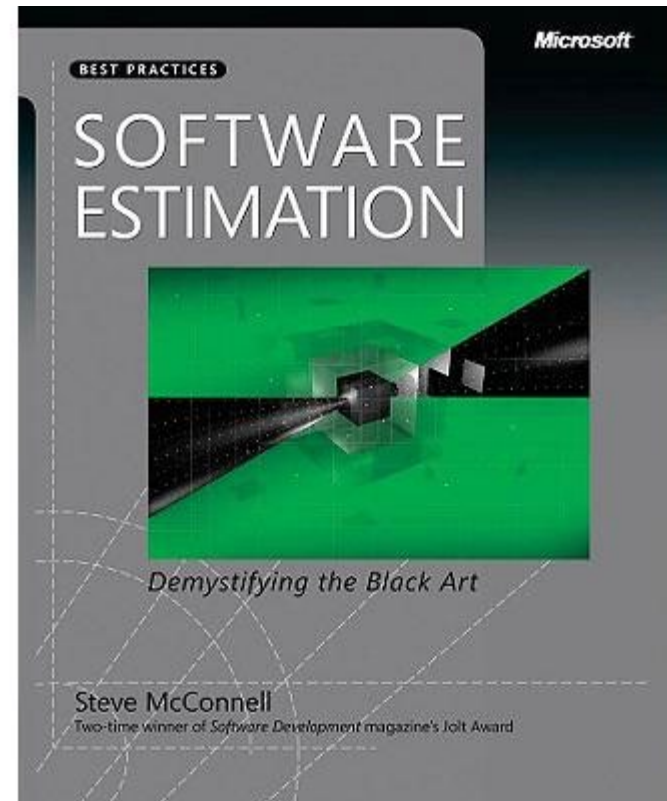
3.2 与用户沟通获取需求的方法——估计

- **目标**：表明一个希望达到的状态。
- **估计**：以当前了解的情况和掌握的资源，要花费多少人力、物力、时间才能实现
- **决心**：保证在某个时间之前完成预先规定的功能和质量。

3.2 与用户沟通获取需求的方法——估计



3.2 与用户沟通获取需求的方法——估计



《Software Estimation: Demystifying the Black Art》

3.2 与用户沟通获取需求的方法——估计

- 中国陆地边界长度 2.2万多公里
- 非洲人口密度 40.58人/平方公里
- 长江一年的流量 9600多亿立方米
- 现年80岁的中国人，从出生起，说过多少句话

3.2 与用户沟通获取需求的方法——估计

硬件项目的估计

- 有一堆砖头，估计有 X 块， N 个人来搬砖，每人每小时可以搬 M 块，那么大概要搬多少时间？
- 估计大概要 $X/N/M$ 小时。

3.2 与用户沟通获取需求的方法——估计

软件项目的估计

- 估计的需求
- 估计的需求复杂度
- 估计的技术难度
- 估计的人员能力
- 人员流动和不可替代性
-

例：如果你一天，可以写一千行C++代码，10天时间，你能写好一万行代码吗？

3.2 与用户沟通获取需求的方法——估计

“我们其实并不是不会估计，我们真正不会的，是把估计后面藏着的种种假设全部列举出来”

——Paul Rook

一个小组的同学(6-8人)决定要徒步遍历中国陆地边界，假设硬件装备齐全，估计需要多长时间？

Wideband Delphi

3.2 与用户沟通获取需求的方法——估计

Wideband Delphi

- 1. 找到一个主持人
- 2. 主持几轮讨论，先确定大家对目标有统一的理解。
- 3. 然后每一轮统计大家对时间的估计，并且询问大家估计值的前提假设是什么，找到合理的假设，然后继续。

3.2 与用户沟通获取需求的方法——估计

第一轮，大家的估计和假设：

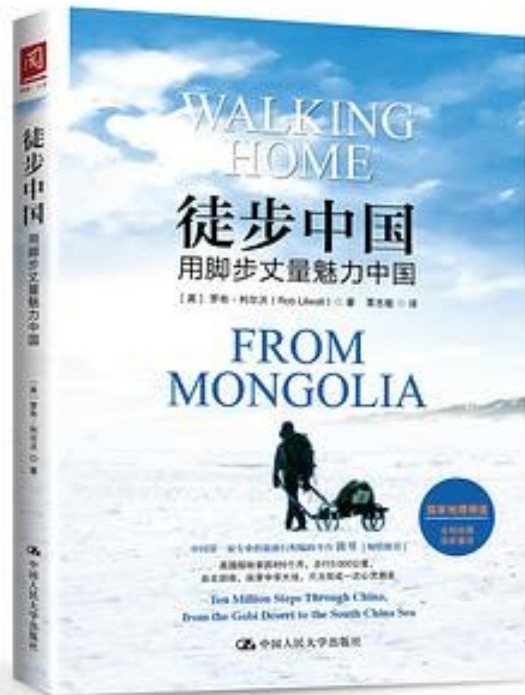
- a. 不可能 假设：徒步团队第二天就解散了。
- b. 300年 假设：要按照边界走，爬到珠穆朗玛峰就挂了。
- c. 5年 假设：3万千米，一年走300天，每天20公里。
- d. 2年 假设：参考长征的记录，12个月走1万2千公里。
- e. 1年 假设：边界大概10个省，在公安边防的带领下完成。
- f. 5个月 假设：沿着南宋时期的边界走走玩玩，很快走完。
- g. 50天 假设：不祥。

3.2 与用户沟通获取需求的方法——估计

新的共同假设:

- 沿着边界 = 离边界最近的公路或小路走，青藏高原可以绕一下，不必亲自到每一块界碑跟前拍照留念。不过还得自己用脚走。
- 陆地边界 = 中国现在的边界，不是南宋时期的。约等于2万千米。

3.2 与用户沟通获取需求的方法——估计



简阳老人198天环游中国 完成2.5万公里行程--四川新闻

2012年11月19日 5月3日,3名平均年龄63岁的老人,骑着自行车从简阳市区出发,预计210天行程2.5万公里,环游中国一圈。(本报5月4日曾作报道)11月17日晚,除了74岁的曾德全老人因身体...



3.2 与用户沟通获取需求的方法——估计

- 另一种估计方法：**经验公式**

$$Y = X \pm (X \div N)$$

Y ——实际时间花费

X ——估计时间

N ——以前做过类似开发工作的次数

3.2 与用户沟通获取需求的方法——估计

- 小明估计开发一个网站的用户管理系统需要3天：

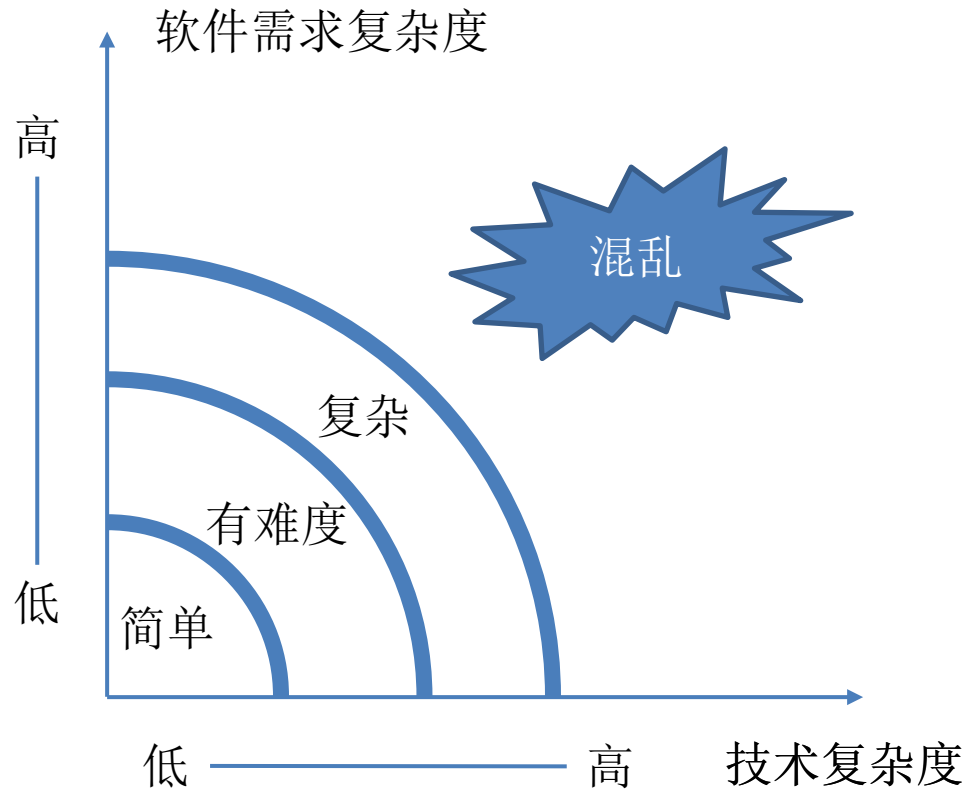
实际开发时间： $Y = 3 \pm (3 \div N)$

- 如果小明从来没有做过用户管理系统， $N=0$
- 如果小明已经做过10次类似的系统， $N=10$

3.2 与用户沟通获取需求的方法——估计

项目的复杂程度：

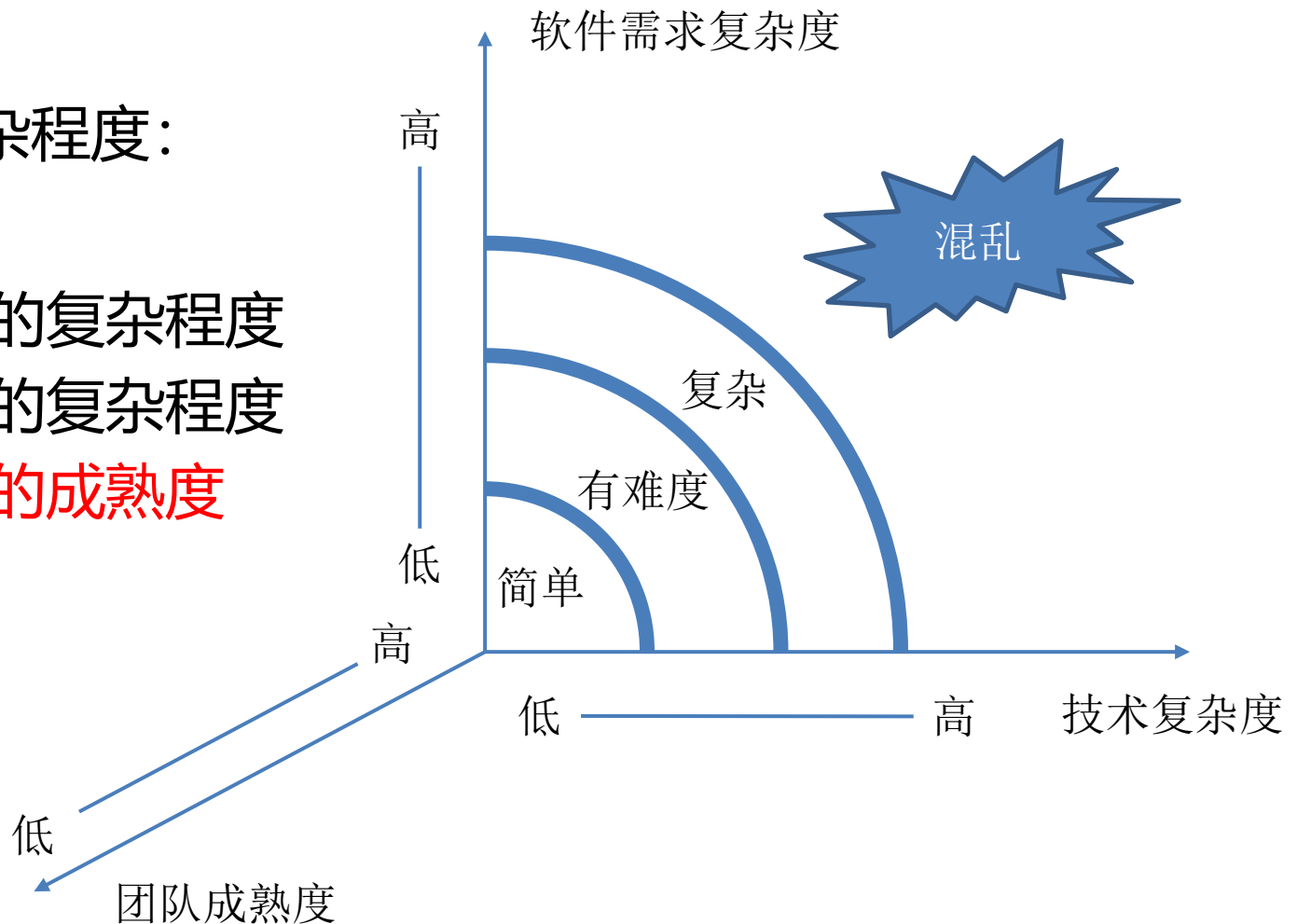
- 1. 需求的复杂程度
- 2. 技术的复杂程度



3.2 与用户沟通获取需求的方法——估计

项目的复杂程度：

- 1. 需求的复杂程度
- 2. 技术的复杂程度
- 3. 团队的成熟度



3.2 与用户沟通获取需求的方法——估计

1. 需求复杂度
2. 技术复杂度
3. 团队成熟度
4. 产品因素
5. 平台因素
6. 人员因素
7. 项目因素



COCOMO 模型



3.2 与用户沟通获取需求的方法——估计

1. 公式项目的复杂程度:

- $(Y_0, Y_0 \times F_0 \times F_1 \times \dots \times F_n)$
 - Y_0 是团队估计的项目时间
 - F_0 到 F_n 是各种因素的值, 取0~10

2. 方法

- 自底向上
- 回溯

3.2 与用户沟通获取需求的方法——估计

敏捷开发中的山寨方法：

- 扑克牌估计法
- 划拳估计法
- T恤尺寸估计法

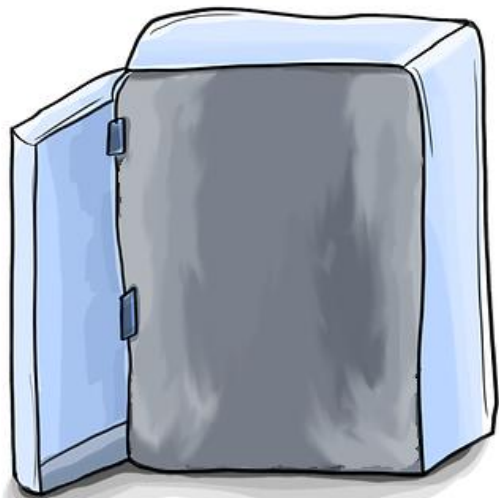


3.2 与用户沟通获取需求的方法——其他方法

1. 竞争性需求分析的框架(NABCD)
2. 功能的定位和优先级（四象限方法）
3. 计划和估计(Wideband Delphi)
4. 分而治之（任务划分技术WBS）

3.2 与用户沟通获取需求的方法——分而治之

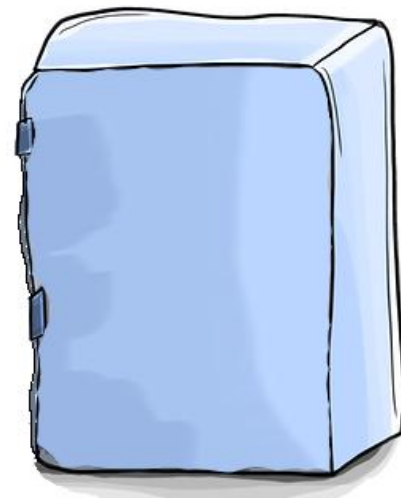
怎么把大象装进冰箱？



I 打开冰箱



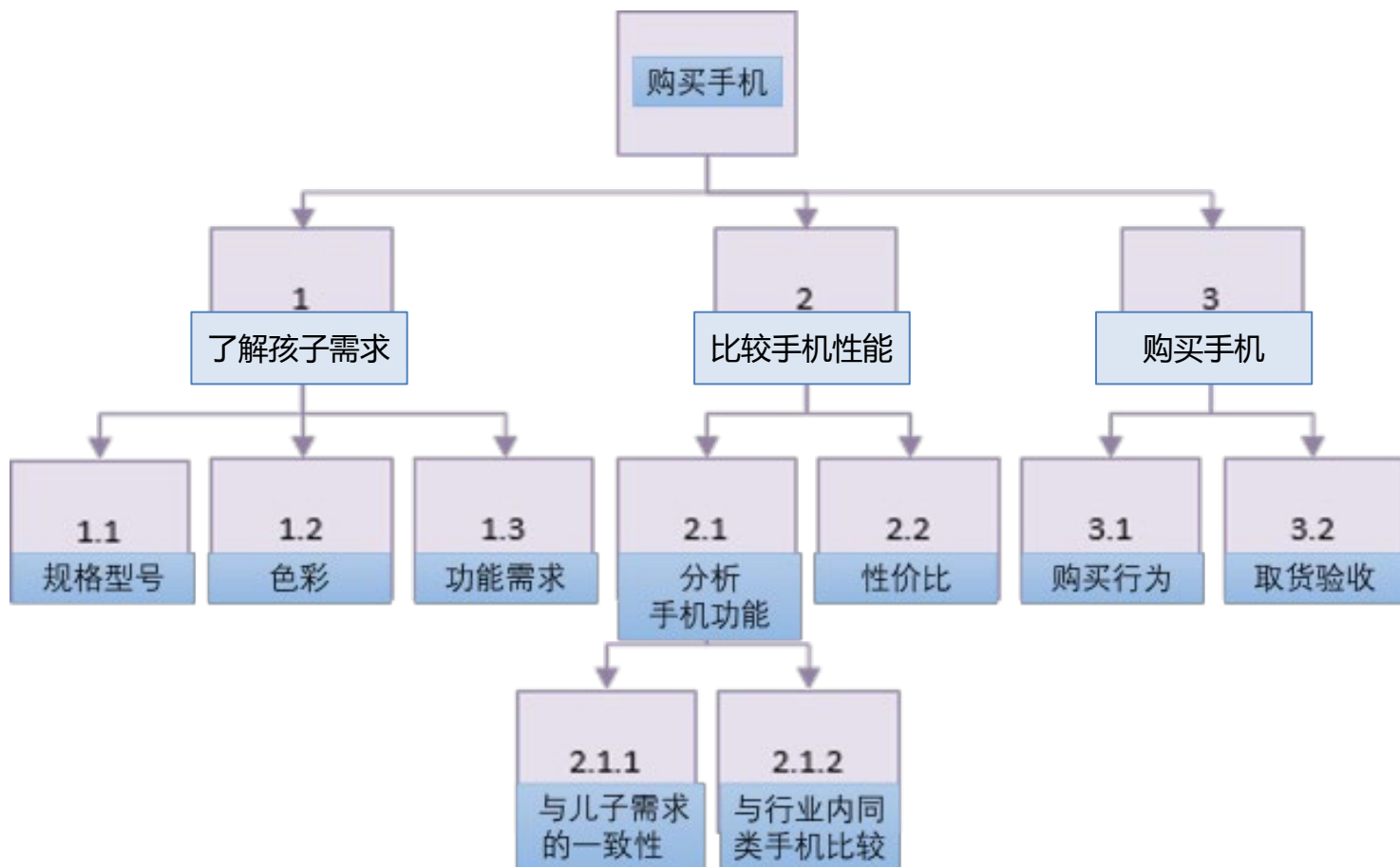
II 把大象塞进去



III 关上冰箱

3.2 与用户沟通获取需求的方法——分而治之

WBS: Work Breakdown Structure



导致需求错误的原因有哪些

- (1) 缺乏足够的用户参与：
 - 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。
 - 可能原因：与用户合作不如编写代码有意思；开发人员觉得已经明白用户的需求了。
- (2) 用户需求不断增加：
 - 在开发过程中，用户需求经常发生变化，不断的变更会使其整体结构越来越乱，整个程序也难以理解和维护。
 - 在项目的开始对项目视图、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。

导致需求错误的原因有哪些

- (3) 需求模棱两可：
 - 不同的人对需求说明产生了不同的理解，或者同一个人能用不止一个方式来解释某项需求说明。
 - 返工耗费开发总费用的40%，而70%~85%的重做是由于需求方面的错误引起的，它是需求规格说明中最严重的问题。
 - 组织不同的人员从不同的角度审查需求。
- (4) 添加不必要的特性：
 - 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，然而常常是用户并不认为这些功能性很有用。
 - 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的需要和允许时限内的技术可行性之间求得平衡。

导致需求错误的原因有哪些

- (5) 规格说明过于简单：
 - 客户往往不明白需求分析的重要性，只是提供一份十分简略的规格说明，仅涉及产品概念上的内容，然后让开发人员在项目进展中去完善，从而导致开发人员先建立产品结构再完成需求说明。
- (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。
 - 在项目早期就针对所有这些主要用户进行分类。

需求分析

3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体-联系图

3.5 数据规范化

3.6 状态转换图+有穷状态机

3.7 其他图形工具

3.8 验证软件需求

3.9 小结

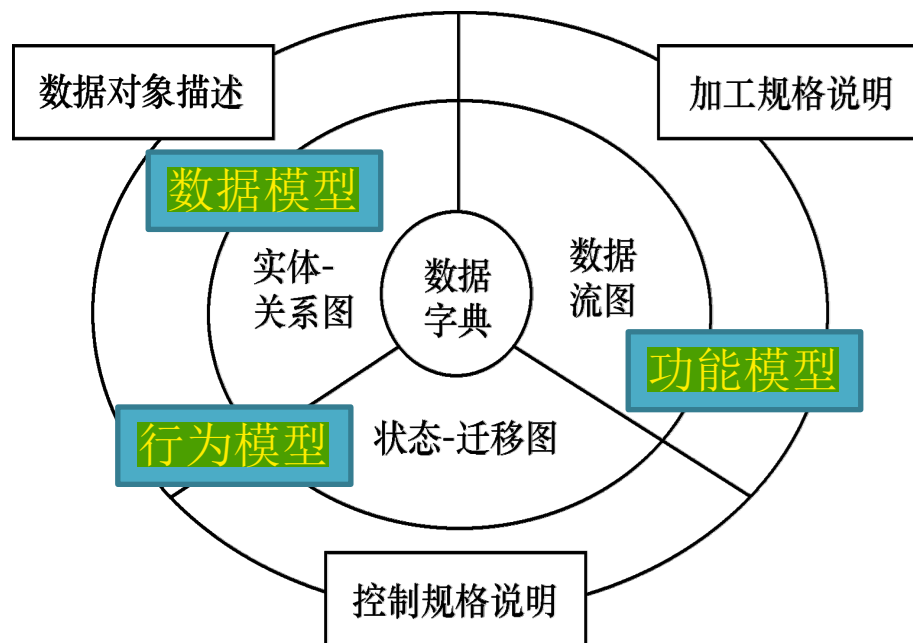
3.3 分析建模与规格说明——分析建模

模型：

——就是为了理解事物而对事物做出的一种抽象，是对事物的一种无歧义的书面描述。通常，由一组图形符号和组织这些符号的规则组成。

3.3 分析建模与规格说明——结构化分析方法

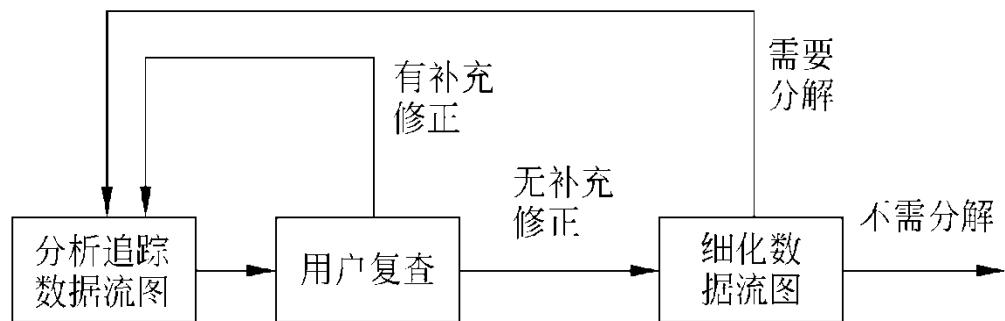
- 传统的分析建模方法称为**结构化分析**（Structured Analysis, SA）方法。
- 结构化分析的模型



3.3 分析建模与规格说明——结构化分析方法

结构化分析方法就是面向数据流自顶向下逐步求精进行需求分析的方法。

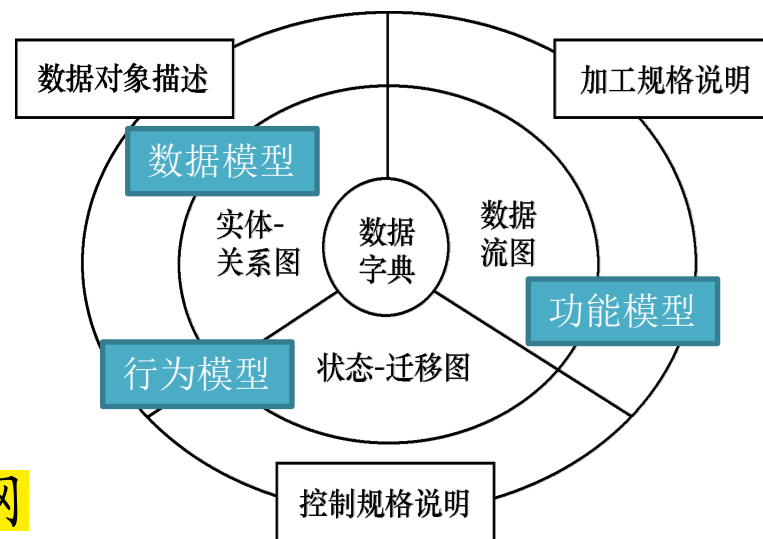
- 1、沿数据流图回溯
- 2、用户复查
- 3、细化数据流图
- 4、修正开发计划
- 5、书写文档
- 6、审查和复审



3.3 分析建模与规格说明——结构化分析方法

➤ 具体的建模方法:

- ◆ 功能模型——数据流图
- ◆ 数据模型——实体关系图
- ◆ 行为模型——状态图、Petri网



3.3 分析建模与规格说明——软件需求规格说明

软件需求规格(Software Requirement Specification)

用自然语言+模型，完整、准确、具体地描述系统的数据要求、功能需求、性能需求、可靠性和可用性要求、出错处理需求、接口需求、约束、逆向需求以及将来可能提出的要求。

软件需求规格说明书，是需求分析阶段得出的最主要的文档。

软件需求规格说明书的编写提示 (GB856T—88)

目 录

1 引言

1.1 编写目的

1.2 背景

1.3 定义

1.4 参考资料

2 任务概述

2.1 目标

2.2 用户的特点

2.3 假定和约束

软件需求规格说明书的编写提示 (GB856T—88)

目 录

3 需求规定

3.1 对功能的规定

3.2 对性能的规定

3.2.1 精度

3.2.2 时间特性要求

3.2.3 灵活性

3.3 输入输出要求

3.4 数据管理能力要求

3.5 故障处理要求

3.6 其他专门要求

4 运行环境规定

4.1 设备

4.2 支持软件

4.3 接口

4.4 控制

3.3 分析建模与规格说明——加工规格说明

- 在对数据流图的分解中，位于层次树最低层的加工也称为基本加工或原子加工，对于每一个基本加工都需要进一步说明，这称为加工规格说明。
- 在编写基本加工的规格说明时，主要目的是要表达“做什么”，而不是“怎样做”。

3.3 分析建模与规格说明——加工规格说明

加工名: 确定能否供货

编号: 1.2

激发条件: 接收到合格订单时

优先级: 普通

输入: 合格订单

输出: 可供货订单

缺货订单加工逻辑: 根据库存记录

IF 订单项目的数量 < 该项目库存量的临界值

THEN 可供货处理

ELSE 此订单缺货, 登录, 待进货后再处理

ENDIF

3.3 分析建模与规格说明——加工规格说明

- 加工规格说明应满足如下的要求：

- (1) 对数据流图的每一个基本加工，必须有一个加工规格说明。
- (2) 加工规格说明必须描述基本加工如何把输入数据流变换为输出数据流的加工规则。
- (3) 加工规格说明必须描述实现加工的策略而不是实现加工的细节。
- (4) 加工规格说明中包含的信息应是充足的，完备的，有用的，没有重复的多余信息。

3.3 分析建模与规格说明——决策表

- **决策表：** 由4个部分组成：
 - 左上部分是**条件茬**，在此区域列出了各种可能的单个条件；
 - 左下部分是**动作茬**，在此区域列出了可能采取的单个动作；
 - 右上部分是**条件项**，在此区域列出了针对各种条件的每一组条件取值的组合；
 - 右下部分是**动作项**，这些动作项与条件项紧密相关，它指出了在条件项的各组取值的组合情况下应采取的动作。

3.3 分析建模与规格说明——决策表

- 决策表举例：商店业务处理系统中检查订货单的决策表。

		1	2	3	4
条件	订货单金额	>5000元	>5000元	≤5000元	≤5000元
	偿还欠款情况	>60天	≤60天	>60天	≤60天
动作	在偿还欠款前不予批准	√			
	发出批准书		√	√	√
	发出发货单		√	√	√
	发催款通知书			√	

3.3 分析建模与规格说明——决策表

• 建立决策表的步骤

1. 列出与一个具体过程（或模块）有关的所有处理。
2. 列出过程执行期间的所有条件（或所有判断）。
3. 将特定条件取值组合与特定的处理相匹配，消去不可能发生的条件取值组合。
4. 将右部每一纵列规定为一个处理规则，即对于某一条件取值组合将有什么动作。

		1	2	3	4
条件	订货单金额	>5000元	>5000元	≤5000元	≤5000元
	偿还欠款情况	>60天	≤60天	>60天	≤60天
动作	在偿还欠款前不予批准	√			
	发出批准书		√	√	√
	发出发货单		√	√	√
	发催款通知书			√	

3.3 分析建模与规格说明——决策表

- 决策表的改进

如果表中有两条或更多的处理规则具有相同的动作，并且其条件项之间存在着某种关系，可以设法将它们合并。

		1	2	3	4
条件	订货单金额	>5000元	>5000元	≤5000元	≤5000元
	偿还欠款情况	>60天	≤60天	>60天	≤60天
动作	在偿还欠款前不予批准	√			
	发出批准书		√	√	√
	发出发货单		√	√	√
	发催款通知书			√	

3.3 分析建模与规格说明——决策表

- 决策表的改进

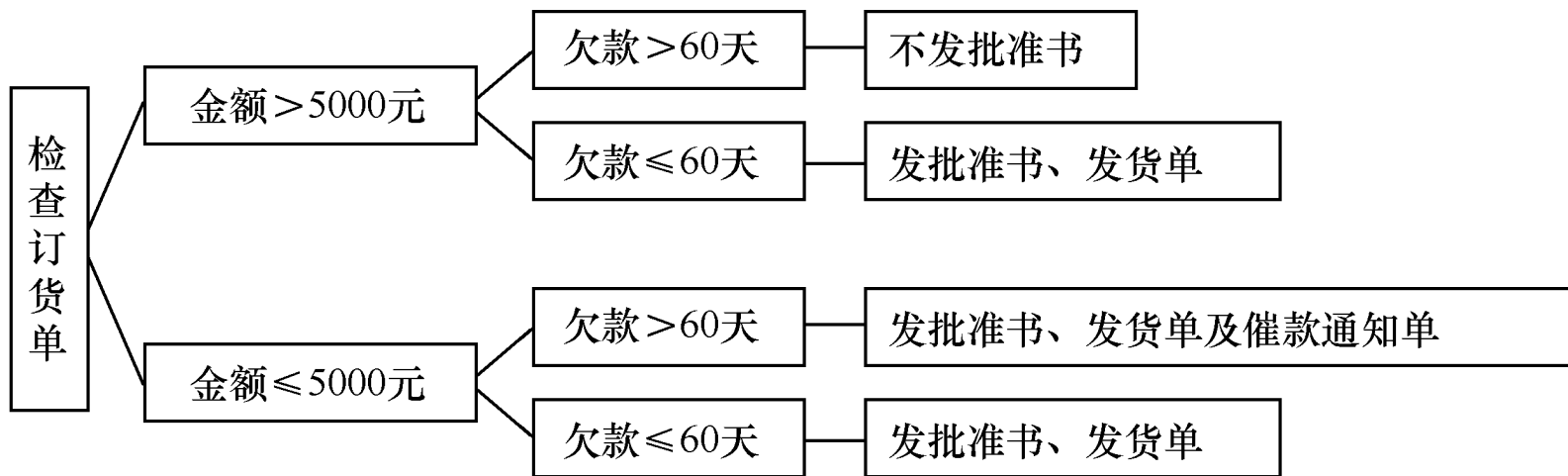
如果表中会有两条或更多的处理规则具有相同的动作，并且其条件项之间存在着某种关系，可以设法将它们合并。

		1	2	3
条件	订货单金额	>5000元	-	≤5000元
	偿还欠款情况	>60天	≤60天	>60天
动作	在偿还欠款前不予批准	√		
	发出批准书		√	√
	发出发货单		√	√
	发催款通知书			√

3.3 分析建模与规格说明——决策树

• 决策树

- 决策树 (Decision Tree) 也是用来表达加工逻辑的一种工具, 有时候它比决策表更直观。
- 检查订货单的决策树



需求分析

3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体-联系图

3.5 数据规范化

3.6 状态转换图

3.7 其他图形工具

3.8 验证软件需求

3.9 小结

3.4 实体-联系图(ER, Entity Relationship Diagram)

- **ER图** ---- 是用来建立数据模型的工具。
- **数据模型** ---- 是一种面向问题的数据模型，是按照用户的观点对数据建立的模型。它描述了从用户角度看到的数据，反映了用户的现实环境，而且与在软件系统中的实现方法无关。
- 数据模型中包含3种相互关联的信息：
 - 数据对象 (**实体**)
 - 数据对象的**属性**
 - 数据对象彼此间相互连接的**关系**

3.4 实体-联系图——(1) 数据对象

- ◆ 数据对象: 是对软件必须理解的复合信息的抽象。
- ◆ 复合信息: 是指具有一系列不同性质或属性的事物, 仅有单个值的事物(例如, 宽度)不是数据对象。
- ◆ 可以由一组属性来定义的实体都可以被认为是数据对象。
如: 外部实体、事物、行为、事件、角色、单位、地点或结构等。
- ◆ 数据对象彼此间是有关联的。

3.4 实体-联系图——(2) 属性

- 属性定义了数据对象的性质。
- 必须把一个或多个属性定义为“标识符”，也就是说，当我们希望找到数据对象的一个实例时，用标识符属性作为“关键字”（通常简称为“键”）。
- 应该根据对所要解决的问题的理解，来确定特定数据对象的一组合适的属性。

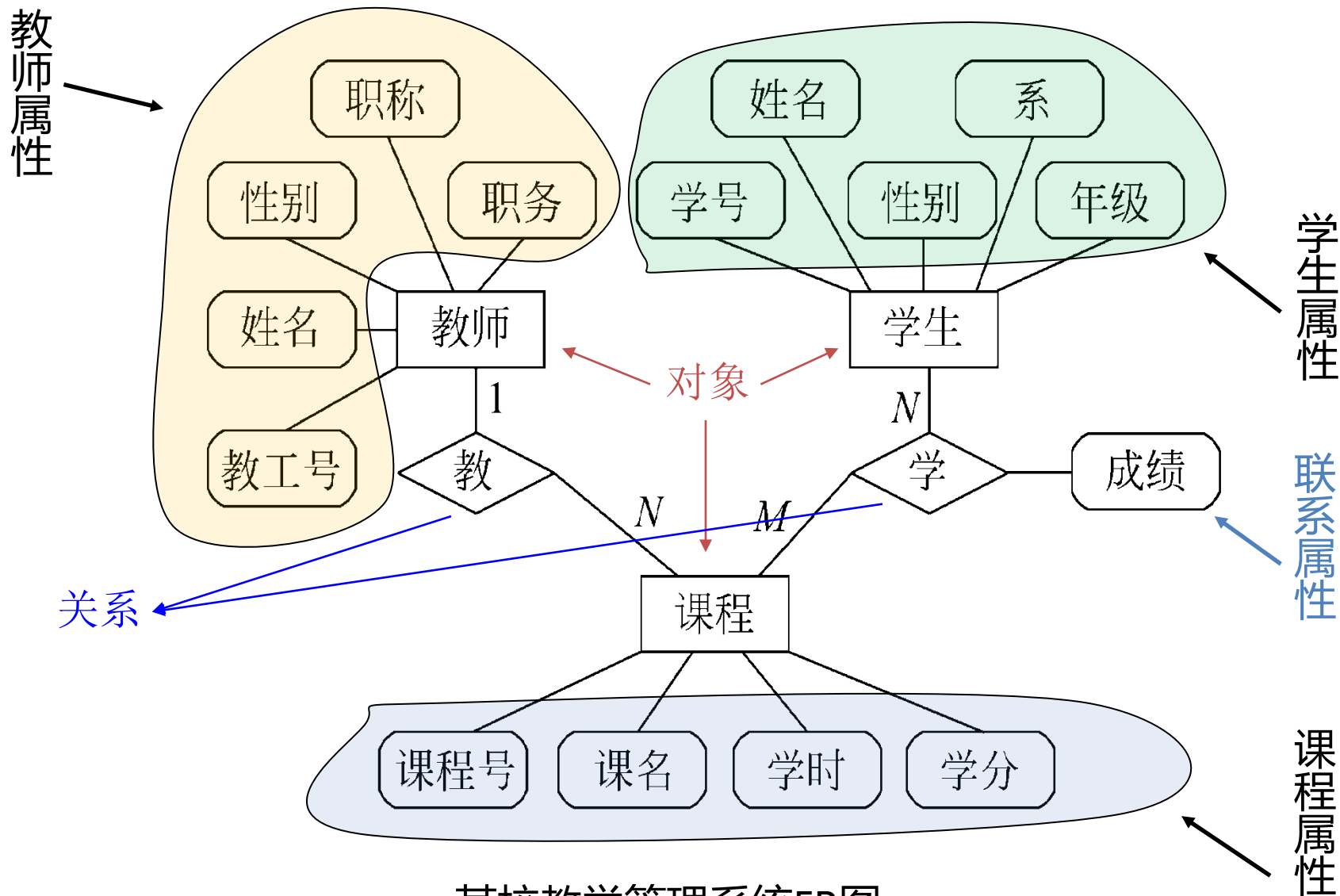
如： 学生具有学号、姓名、性别、年龄、专业（其它略）等属性；
课程具有课程号、课程名、学分、学时数等属性；
教师具有职工号、姓名、年龄、职称等属性。

3.4 实体-联系图——(3) 联系

- ◆ 数据对象彼此之间相互连接的方式称为联系，也称为关系。
- ◆ 联系可分为以下3种类型：
 - a. 一对一联系(1 : 1)
如：一个部门有一个经理，而每个经理只在一个部门任职，则部门与经理的联系是一对一的。
 - b. 一对多联系(1 : N)
如：某校教师与课程之间存在一对多的联系“教”，即每位教师可以教多门课程，但是每门课程只能由一位教师来教。
 - c. 多对多联系(M : N)
如：学生与课程间的联系(“学”)是多对多的，即一个学生可以学多门课程，而每门课程可以有多个学生来学。
- ◆ 联系也可能有属性。
如：学生“学”某门课程所取得的成绩，既不是学生的属性也不是课程的属性。由于“成绩”既依赖于某名特定的学生又依赖于某门特定的课程，所以它是学生与课程之间的联系“学”的属性。

3.4 实体-联系图——(4) 符号

- ◆ ER图中包含实体、关系和属性3种基本成分。
- ◆ 通常用矩形框代表实体；
- ◆ 用连接相关实体的菱形框表示关系；
- ◆ 用椭圆形或圆角矩形表示实体(或关系)的属性；
- ◆ 并用直线把实体(或关系)与其属性连接起来。



某校教学管理系统ER图

需求分析

3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体-联系图

3.5 数据规范化

3.6 状态转换图

3.7 其他图形工具

3.8 验证软件需求

3.9 小结

3.5 数据规范化



为什么数据要规范化？

3.5 数据规范化

规范化的目的是：

1. **消除数据冗余**，即消除表格中数据的重复；
2. **消除多义性**，使关系中的属性含义清楚、单一；
3. **使关系的“概念”单一化**，让每个数据项只是一个简单的数或字符串，而不是一个组项或重复组；
4. **方便操作**。使数据的插入、删除与修改操作可行并方便；
5. **使关系模式更灵活**，易于实现接近自然语言的查询方式。

3.5 数据规范化

学校要聘请名誉校长，如果马斯克、拜登、苏炳添、周深、阿黛尔五人来竞选，请班里十位同学来投票。

名誉校长候选人选票				
候选人名单：				
马斯克	拜登	苏炳添	周深	阿黛尔
请在认为合适的人选姓名下方相应栏内打“v”				
投票人情况：				
姓名：				
所在班级：				
性别：				

3.5 数据规范化

投票人姓名	性别	所在专业	候选人
丁宇庭	男	计科	苏炳添
孙中川	男	计科	周深
倪慧明	女	计科	马斯克
高原	男	计科	苏炳添
白钰	女	计科	阿黛尔
吴丽雅	女	计科	马斯克
杜鹏俊	男	计科	拜登
袁航	男	计科	周深
栾志超	男	计科	周深
蒋蓉	女	计科	马斯克

3.5 数据规范化

如何规范化?

➤ --- 将数据的逻辑结构归结为满足一定条件的二维表（关系）：

1. 表格中每个信息项必须是一个不可分割的数据项，不可是组项。
2. 表格中每一列（列表示属性）中所有信息项必须是同一类型，各列的名字（属性名）互异，列的次序任意。
3. 表格中各行（行表示元组）互不相同，行的次序任意。

教工号	姓名	性别	职称	职务
001	张毅坤	男	教授	院长
002	李 林	女	讲师	

3.5 数据规范化——案例

案例(教学管理系统):

1. 有三个实体型，即课程、学生和教师，用三个关系保存它们的信息：

学生(学号，姓名，性别，年龄，年级，专业，籍贯)

教师(职工号，姓名，年龄，职称，职务，工资级别，工资)

课程(课程号，课程名，学分，学时，课程类型)

2. 为表示实体型之间的联系，又建立两个关系：

选课(学号，课程号，听课出勤率，作业完成率，分数)

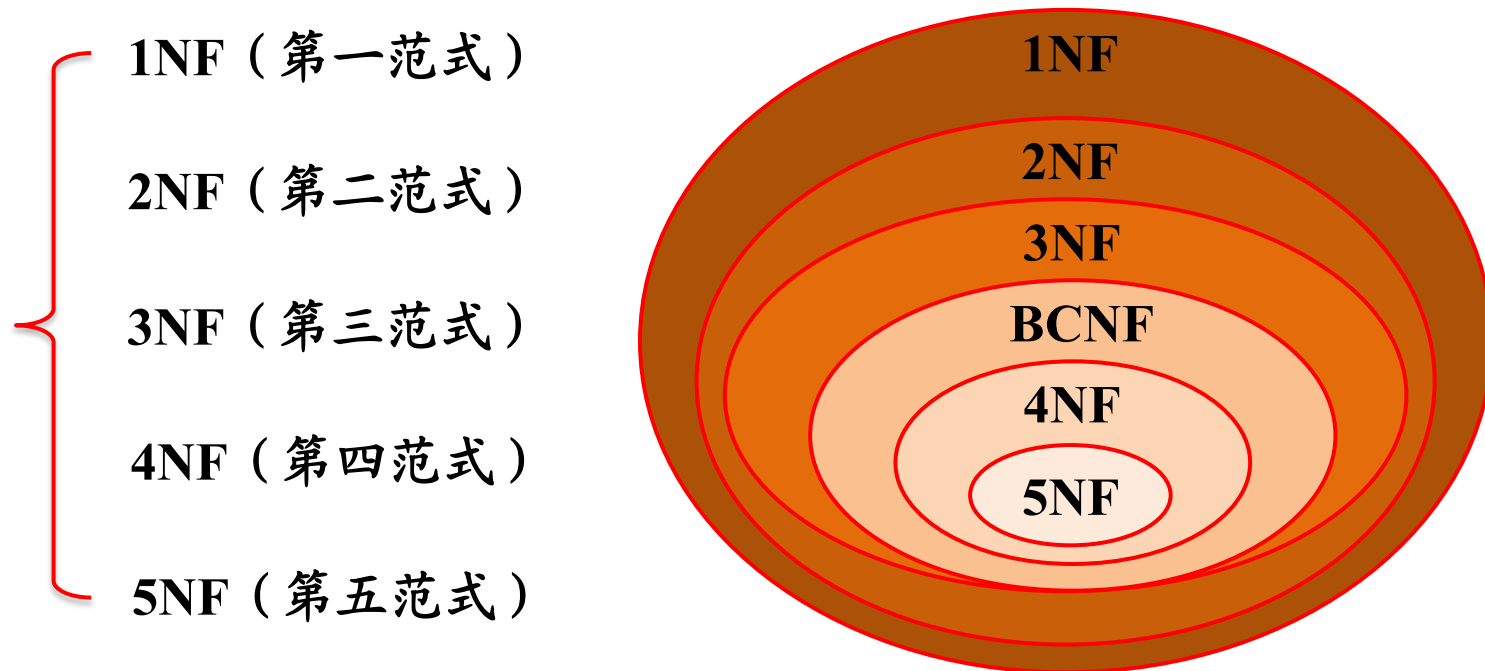
教课(职工号，课程号，授课效果)

3. 这五个关系，组成了数据库的模型。

4. 在每个关系中，属性名下加（下划线）指明关键字。并规定关键字能唯一地标识一个元组。

3.5 数据规范化——范式

消除数据冗余：



通常用“**范式(Normal Forms)**”定义消除数据冗余的程度。**第一范式(1 NF)**数据冗余程度最大，**第五范式(5 NF)**数据冗余程度最小。

3.5 数据规范化——范式

通常用“**范式(Normal Forms)**”定义**消除数据冗余的程度**。第一范式(1 NF)数据冗余程度最大，第五范式(5 NF)数据冗余程度最小。

但是：

- 1、范式级别越高，存储同样数据就需要**分解成更多张表**，因此，“存储自身”的过程也就越复杂。
- 2、随着范式级别的提高，数据的**存储结构与基于问题域的结构间的匹配程度也随之下降**，因此，在**需求变化时数据的稳定性较差**。
- 3、范式级别提高则需要**访问的表增多**，因此**性能(速度)将下降**。

3.5 数据规范化——非第一范式

学号	姓名	专业		课程名称	分数
		专业名称	专业带头人		
10010	张无忌	经济	张三丰	高等数学	95
10010	张无忌	经济	张三丰	大学英语	87
10010	张无忌	经济	张三丰	计算机基础	65
10011	令狐冲	法律	任我行	法理学	77
10011	令狐冲	法律	任我行	刑法	90
10011	令狐冲	法律	任我行	法律社会学	65
10012	杨过	法律	任我行	法理学	95
10012	杨过	法律	任我行	刑法	87
10012	杨过	法律	任我行	法律社会学	75

第一范式：每一列都是不可分割的原子数据项

3.5 数据规范化——第一范式

学号	姓名	专业名称	专业带头人	课程名称	分数
10010	张无忌	经济	张三丰	高等数学	95
10010	张无忌	经济	张三丰	大学英语	87
10010	张无忌	经济	张三丰	计算机基础	65
10011	令狐冲	法律	任我行	法理学	77
10011	令狐冲	法律	任我行	刑法	90
10011	令狐冲	法律	任我行	法律社会学	65
10012	杨过	法律	任我行	法理学	95
10012	杨过	法律	任我行	刑法	87
10012	杨过	法律	任我行	法律社会学	75

第一范式：每一列都是不可分割的原子数据项

3.5 数据规范化——第二范式

选课表

学号	课程名称	分数
10010	高等数学	95
10010	大学英语	87
10010	计算机基础	65
10011	法理学	77
10011	刑法	90
10011	法律社会学	65
10012	法理学	95
10012	刑法	87
10012	法律社会学	75

学生表

学号	姓名	专业名称	专业带头人
10010	张无忌	经济	张三丰
10011	令狐冲	法律	任我行
10012	杨过	法律	任我行

存在的问题：

- 数据添加存在问题：添加新开设的专业和专业带头人时，数据不合法。
- 数据删除存在问题：张无忌同学毕业了，删除数据后，会将专业数据一起删除。

第二范式：在1NF基础上，非主属性必须完全依赖于主属性

3.5 数据规范化——第三范式

选课表

学号	课程名称	分数
10010	高等数学	95
10010	大学英语	87
10010	计算机基础	65
10011	法理学	77
10011	刑法	90
10011	法律社会学	65
10012	法理学	95
10012	刑法	87
10012	法律社会学	75

学生表

学号	姓名	专业名称
10010	张无忌	经济
10011	令狐冲	法律
10012	杨过	法律

专业表

专业名称	专业带头人
经济	张三丰
法律	任我行
计算机	郭靖

第三范式：在2NF的基础上，任何的非主属性不依赖于其他非主属性

需求分析

3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体-联系图

3.5 数据规范化

3.6 状态转换图

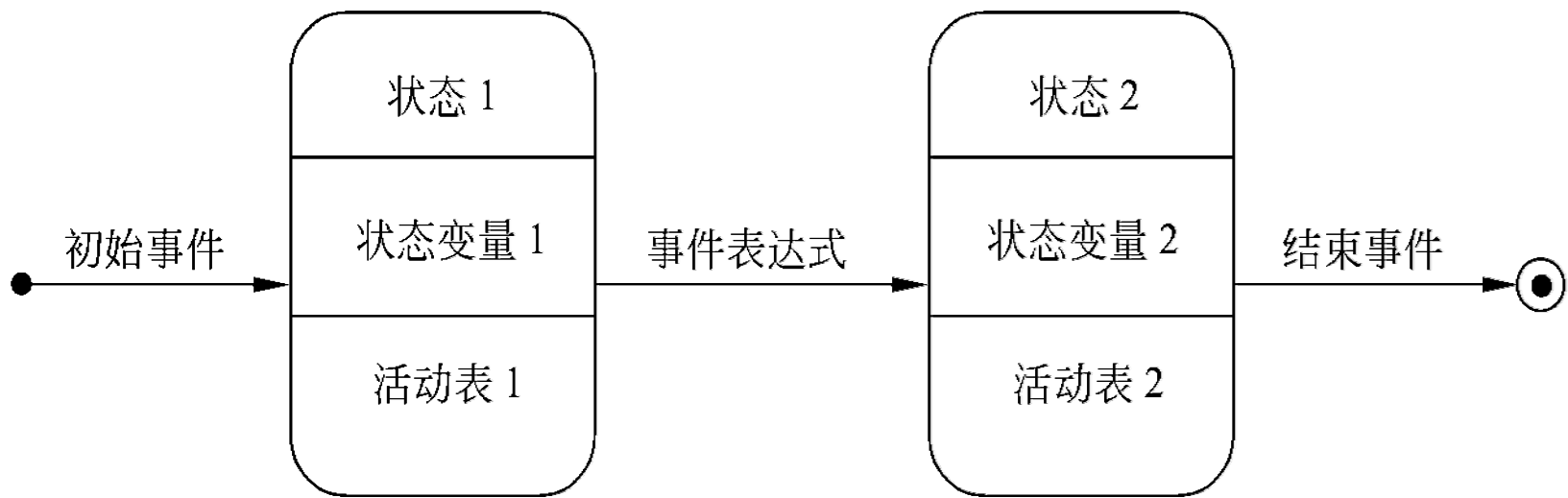
3.7 其他图形工具

3.8 验证软件需求

3.9 小结

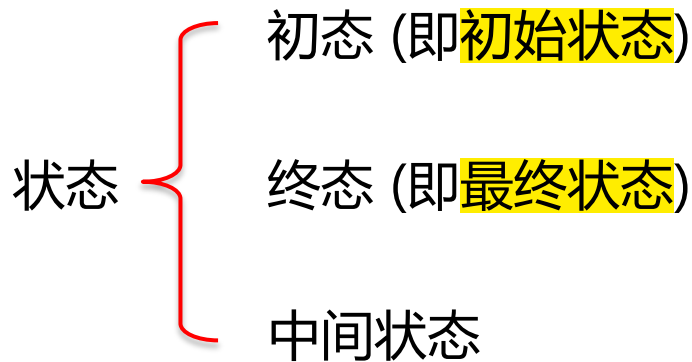
3.6 状态转换图

定义： 通过描绘系统的**状态**及引起系统状态转换的**事件**，来表示系统的**行为**。此外，状态图还指明了作为特定事件的结果系统将做哪些动作(例如，处理数据)。



3.6 状态转换图

状态：状态是任何可以被观察到的系统行为模式，一个状态代表系统的一种行为模式。状态规定了系统对事件的响应方式。系统对事件的响应，既可以是做一个(或一系列)动作，也可以是仅仅改变系统本身的状态，还可以是既改变状态又做动作。



一张状态图中只能有一个初态，而终态则可以有0至多个。

3.6 状态转换图

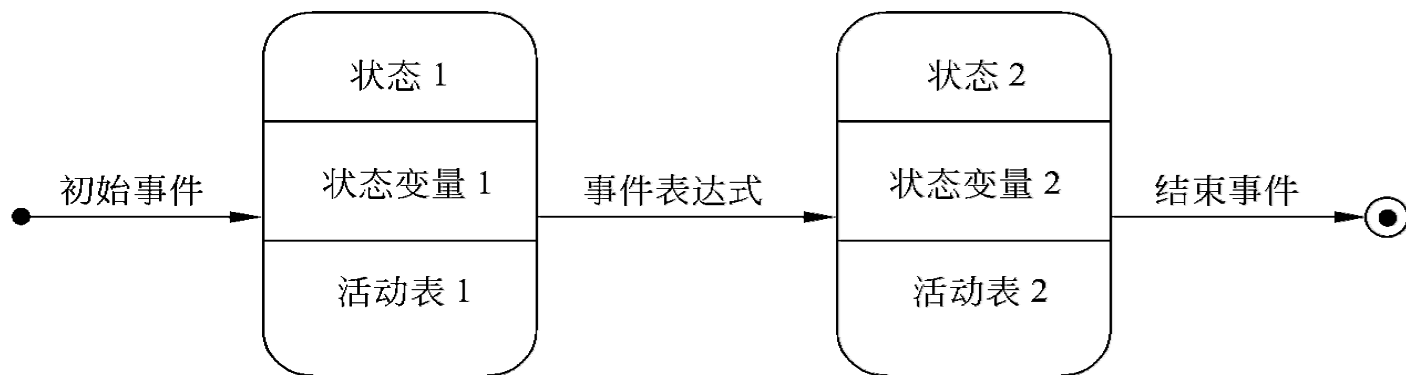
事件：事件是在某个特定时刻发生的事情，它是对引起系统做动作或(和)从一个状态转换到另一个状态的外界事件的抽象。

例如： 内部时钟表明某个规定的时间段已经过去
用户移动或点击鼠标

事件就是引起系统做动作或(和)转换状态的控制信息。

3.6 状态转换图

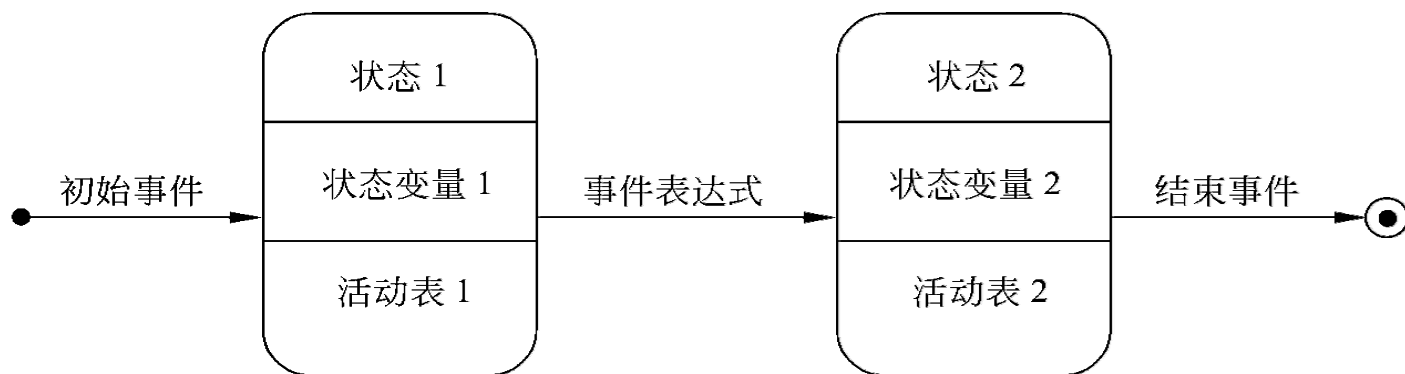
符号：



- 初态用实心圆表示，终态用一对同心圆(内圆为实心圆)表示。
- 中间状态用圆角矩形表示，用两条水平线把它分成上、中、下3个部分：
 - 上面部分为状态的名称，这部分是必须有的；
 - 中间部分为状态变量的名字和值，这部分是可选的；
 - 下面部分是活动表，这部分也是可选的。

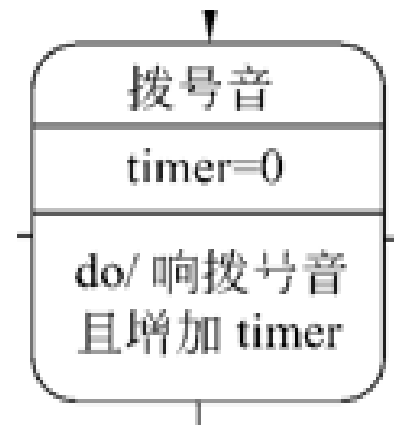
3.6 状态转换图

符号：



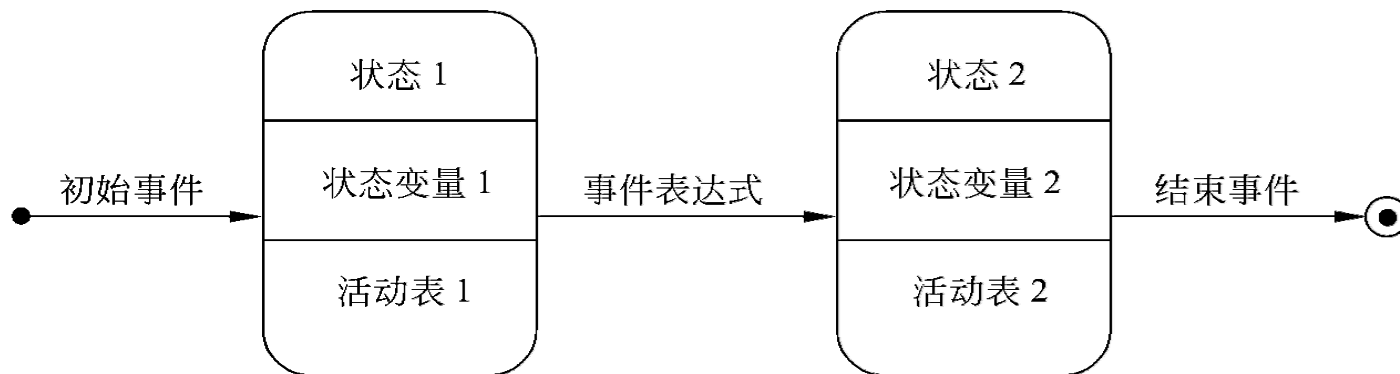
活动表的语法格式：事件名(参数表)/动作表达式

“事件名”可以是任何事件的名称。在活动表中经常使用下述3种标准事件：entry, exit, do
可以为事件指定参数表，活动表中的动作表达式描述应做的具体动作。



3.6 状态转换图

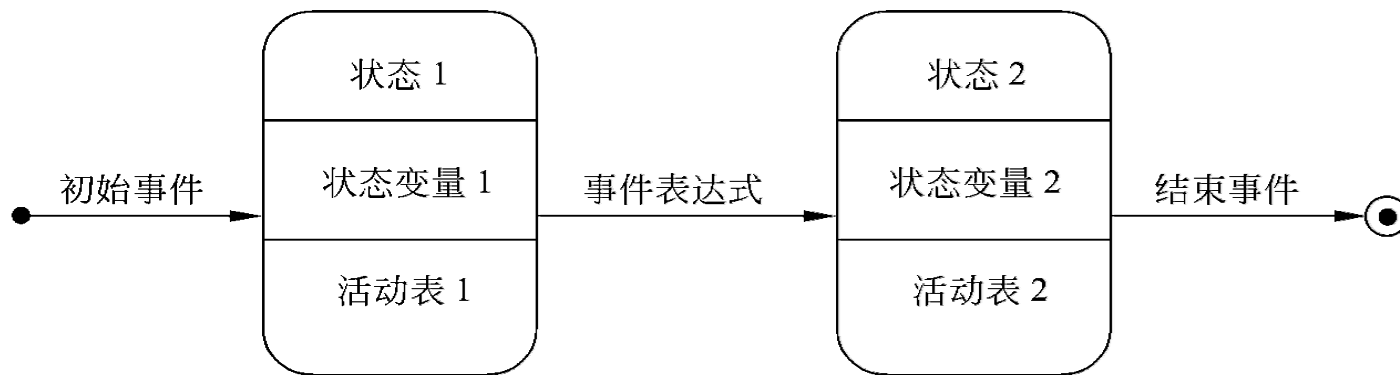
符号：



- 状态之间带箭头的连线称为**状态转换**，箭头指明了转换方向。
- 状态变迁通常是由事件触发的，应在表示状态转换的**箭头线上标出触发转换的事件表达式**；如果在箭头线上未标明事件，则表示**在源状态的内部活动执行完之后自动触发转换**。

3.6 状态转换图

符号：

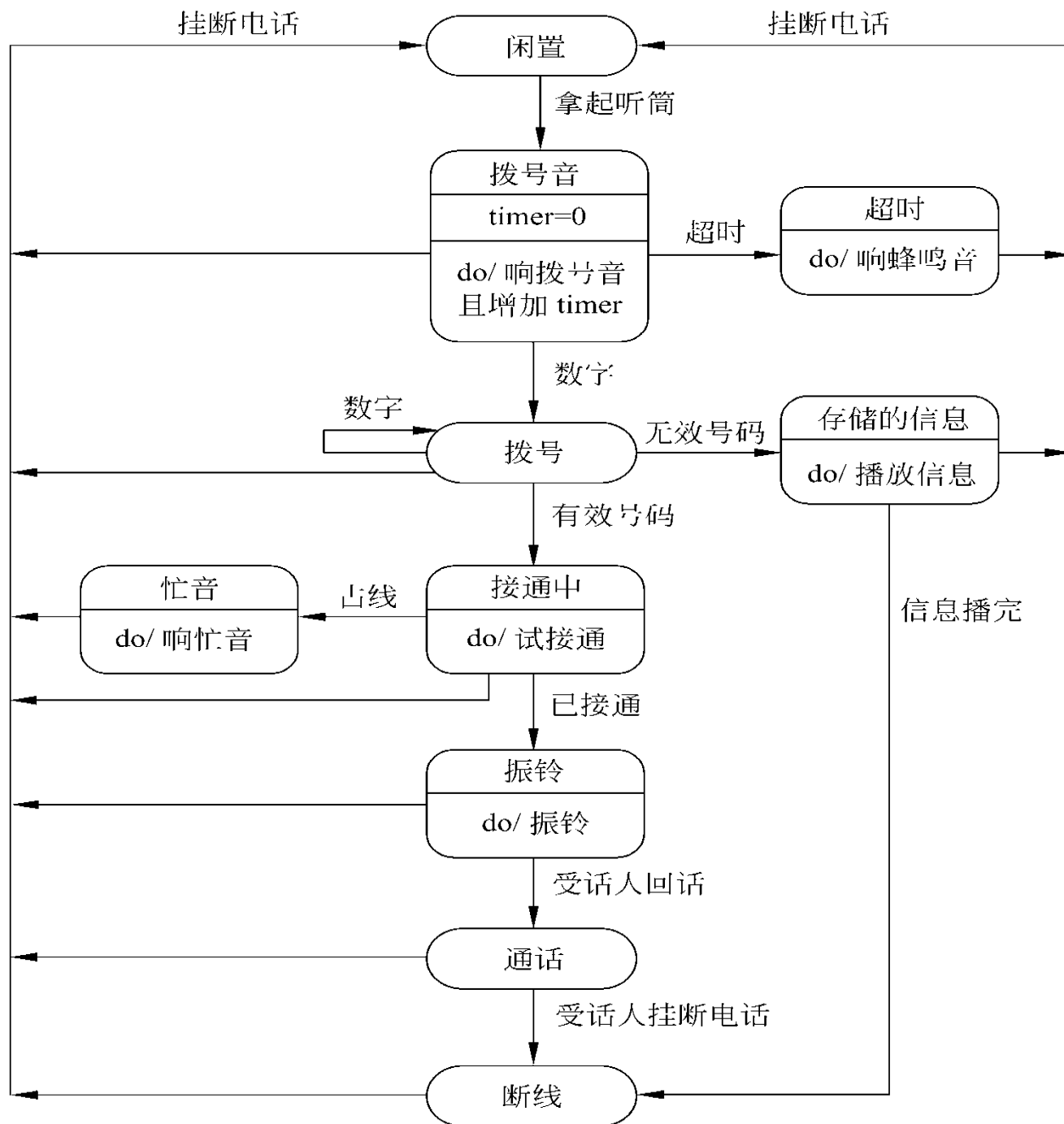


➤ 事件表达式的语法：

事件说明 [守卫条件] / 动作表达式

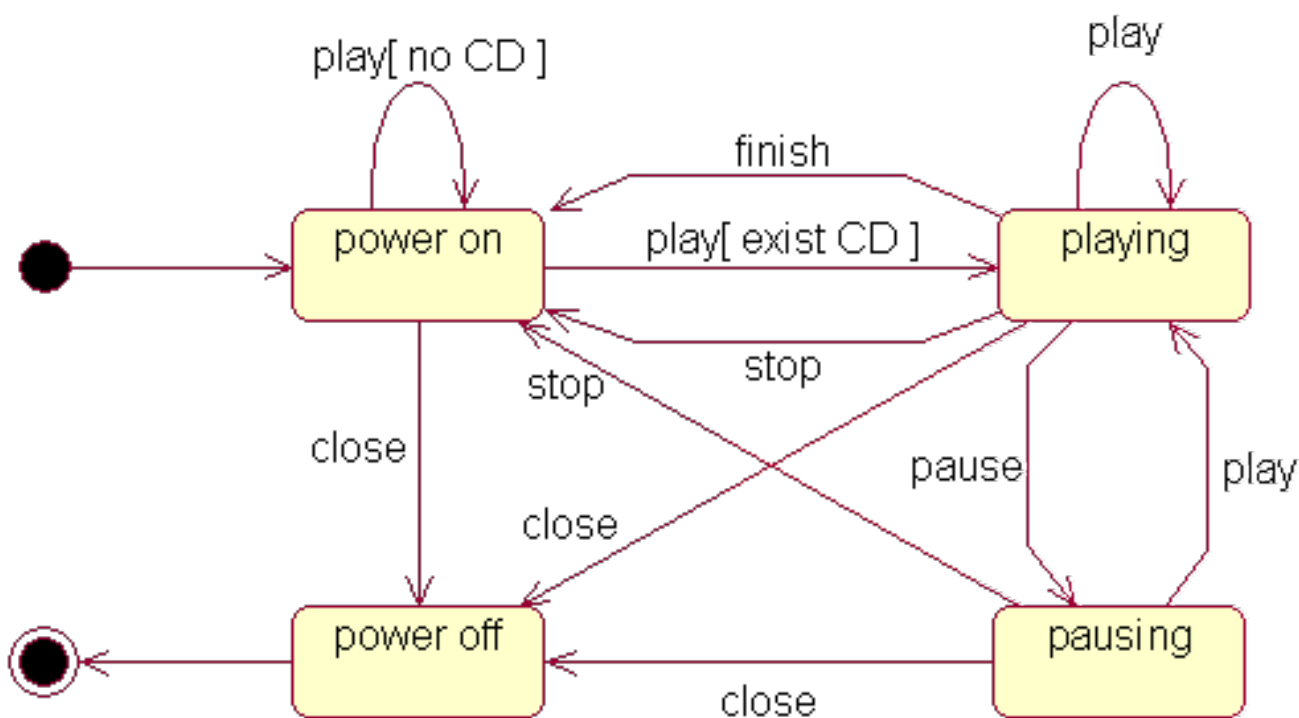
- 事件说明的语法为：事件名(参数表)。
- 守卫条件是一个布尔表达式。如果同时使用事件说明和守卫条件，则当且仅当事件发生且布尔表达式为真时，状态转换才发生。如果只有守卫条件没有事件说明，则只要守卫条件为真状态转换就发生。
- 动作表达式是一个过程表达式，当状态转换开始时执行该表达式。

案例1:



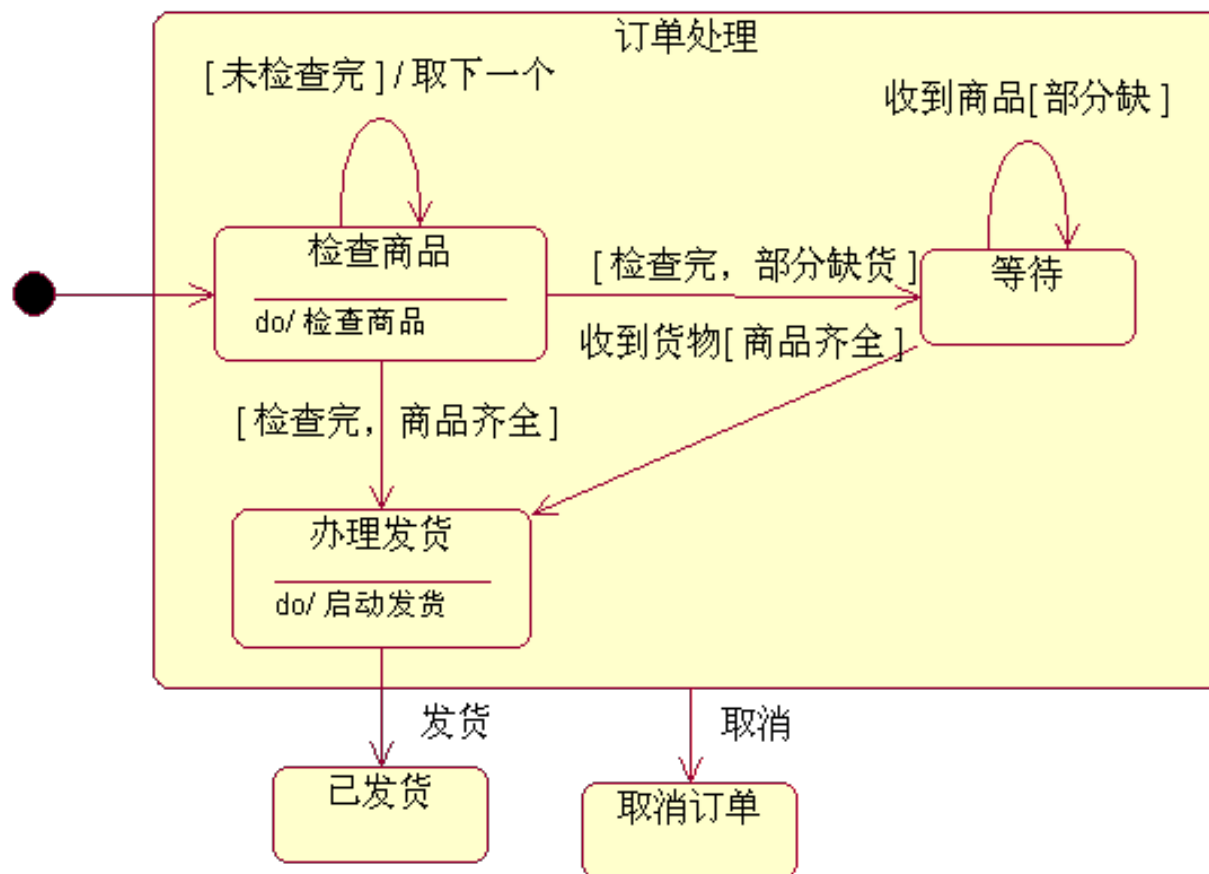
3.6 状态转换图

案例2:



3.6 状态转换图

案例3:



需求分析

3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体-联系图

3.5 数据规范化

3.6 状态转换图

3.7 其他图形工具

3.8 验证软件需求

3.9 小结

3.7 其他图形工具

- ◆ 1. 层次方框图
- ◆ 2. Warnier图
- ◆ 3. IPO图

3.7 其他图形工具——层次方框图

层次方框图

- 层次方框图用树形结构的一系列多层次的矩形框描绘数据的层次结构。
- 树形结构的顶层是一个单独的矩形框，它代表完整的数据结构，下面的各层矩形框代表这个数据的子集，最底层的各个框代表组成这个数据的实际数据元素(不能再分割的元素)。

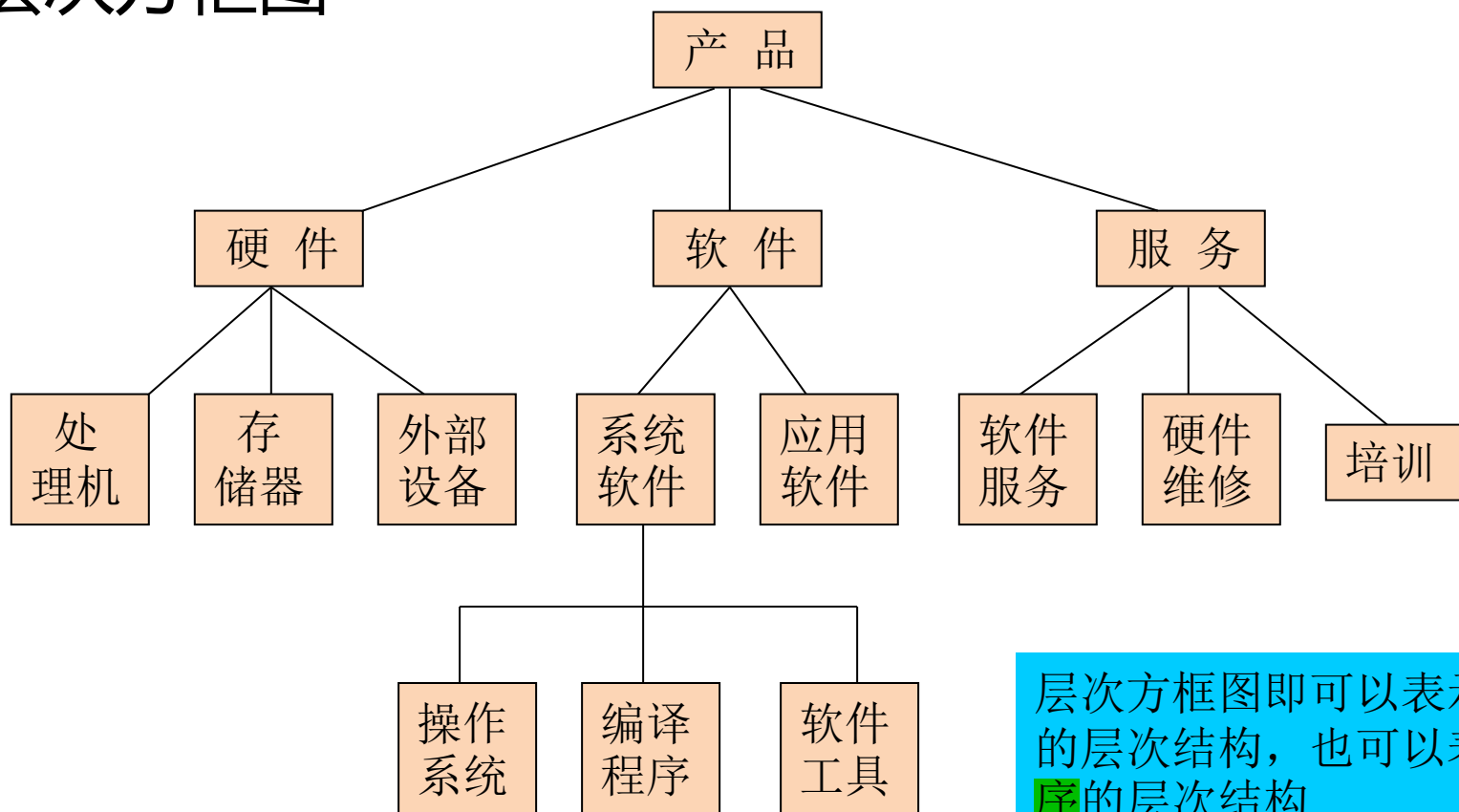
3.7 其他图形工具——层次方框图

层次方框图

- 随着结构的精细化，层次方框图对数据结构也描绘得越来越详细，这种模式非常适合于需求分析阶段的需要。
- 系统分析员从对顶层信息的分类开始，沿图中每条路径反复细化，直到确定了数据结构的全部细节时为止。

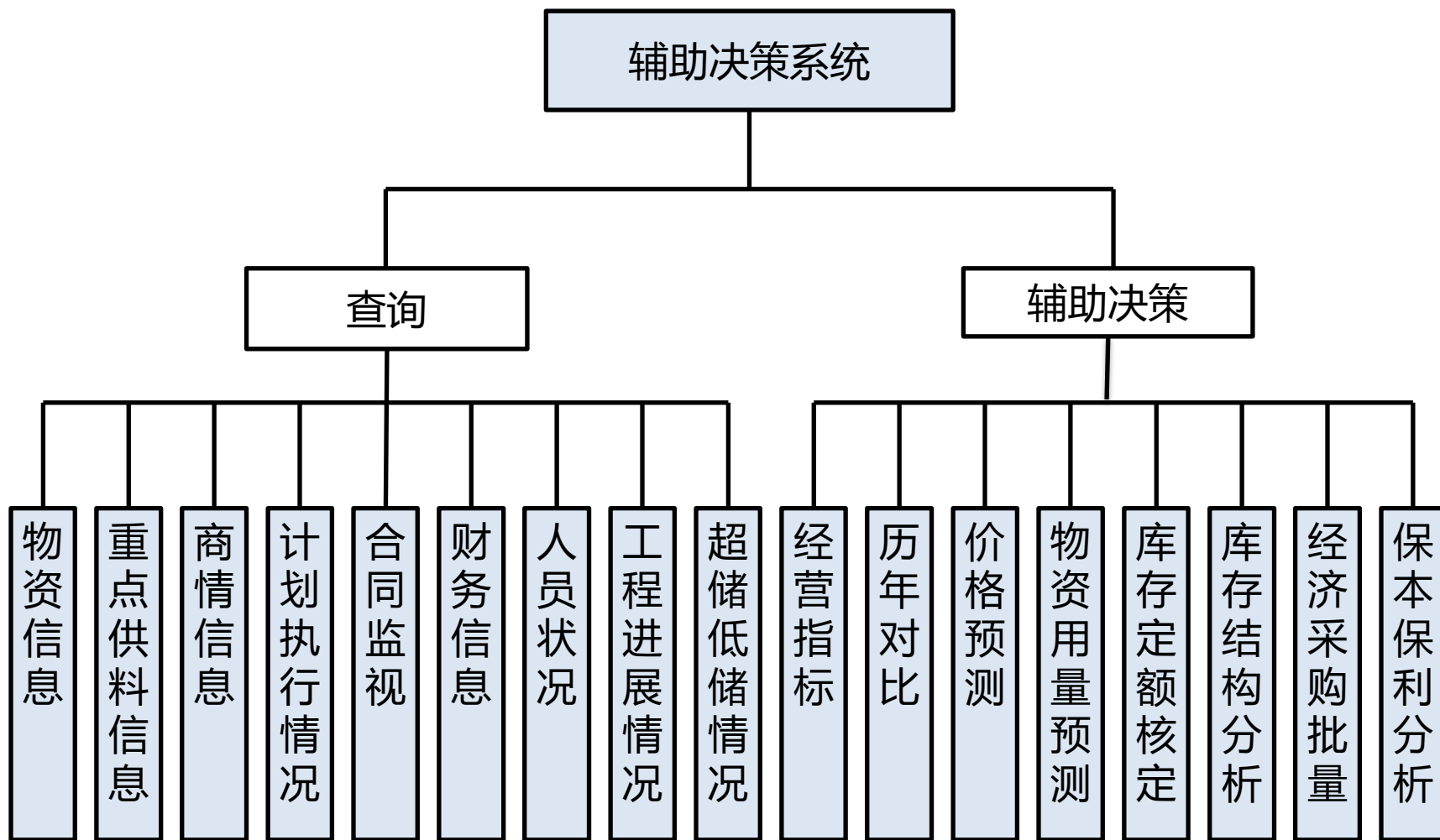
3.7 其他图形工具——层次方框图

层次方框图



层次方框图既可以表示数据的层次结构，也可以表示程序的层次结构

3.7 其他图形工具



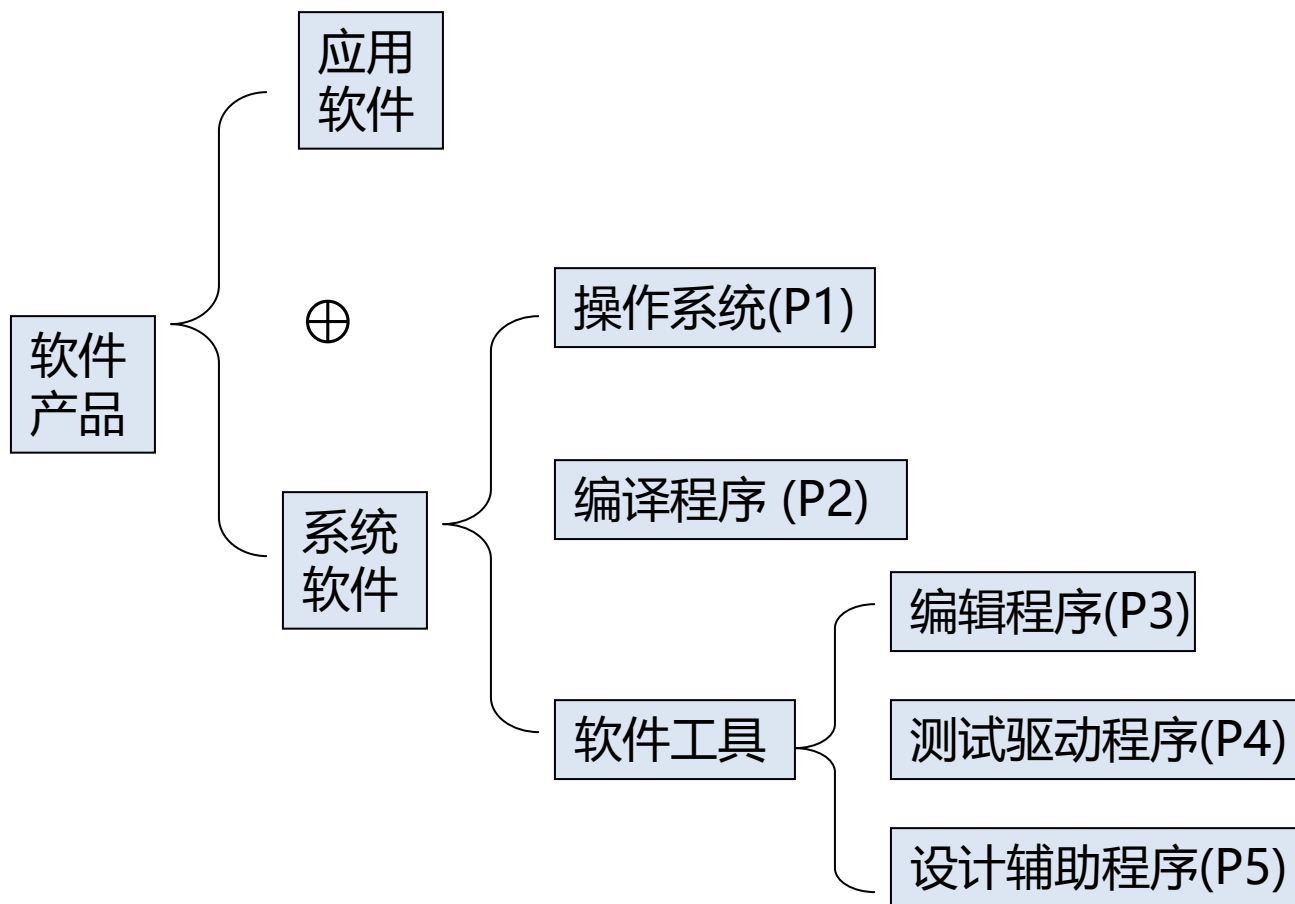
3.7 其他图形工具——Warnier图

Warnier图

- Warnier图也用树形结构描绘信息，这种图形工具比层次方框图提供了更丰富的描述手段，可以表明信息的逻辑组织。
- 它可以指出一类信息或一个信息元素是重复出现的，也可以表示特定信息在某一类信息中是有条件地出现的。
- 重复和条件约束是说明软件处理过程的基础，所以很容易把Warnier图转变成软件设计的工具。

3.7 其他图形工具——Warnier图

Warnier图



3.7 其他图形工具——IPO图

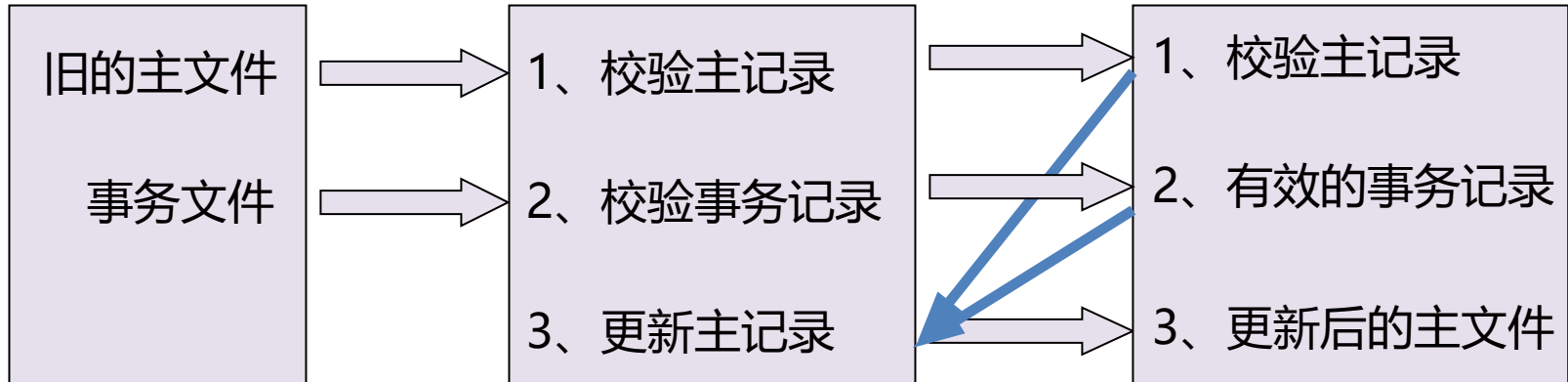
IPO图: 输入/处理/输出图的简称-IBM

- 左边的框中列出有关的输入数据。
- 中间的框内列出主要的处理，处理框中列出处理的次序暗示了执行的顺序，但是用这些基本符号还不足以精确描述执行处理的详细情况。
- 在右边的框内列出产生的输出数据。
- 在IPO图中还用类似向量符号的粗大箭头清楚地指出数据通信的情况。

其他图形工具--IPO图

IPO图: 输入/处理/输出图的简称-IBM

- 类似向量符号的粗大箭头指出数据通信的情况。



- 左边的框中列出有关的输入数据。
- 中间的框内列出主要的处理，处理框中列出处理的执行的顺序，但是用这些基本符号还不足以精确描述执行处理的详细情况。
- 在右边的框内列出产生的输出数据。

3.7 其他图形工具—IPO表

IPO 表	
系统：_____	作者：_____
模块：_____	日期：_____
编号：_____	
被调用：	调用：
输入：	输出：
处理：	
局部数据元素：	注释：

- 在需求分析阶段可以使用IPO表简略地描述系统的主要算法。
- 需求分析阶段，IPO表中的许多附加信息暂时还不具备，但在设计阶段可以进一步补充修正这些图，作为设计阶段的文档。
- 这也是在需求分析阶段用IPO表作为描述算法的工具的重要优点。

需求分析

3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体-联系图

3.5 数据规范化

3.6 状态转换图

3.7 其他图形工具

3.8 验证软件需求

3.9 小结

验证软件需求

验证软件需求的正确性，一般应从4个方面进行：

- (1) **一致性** 所有需求必须是一致的，任何一条需求不能和其他需求互相矛盾。
- (2) **完整性** 需求必须是完整的，规格说明书应该包括用户需要的每一个功能或性能。
- (3) **现实性** 指定的需求应该是用现有的硬件技术和软件技术基本上可以实现的。
- (4) **有效性** 必须证明需求是正确有效的，确实能解决用户面对的问题。

验证软件需求

验证软件需求的方法：

(1) 验证需求的一致性：人工审查→**形式化描述**软件需求，软件工具自动验证。

(2) 验证需求的现实性：参考以往的开发经验，分析，仿真或模拟

(3) 验证需求的完整性和一致性：****原型系统

需求管理工具

- 以数据库为核心的产品（如Caliber-RM和DOORS）：

将所有的需求、属性和跟踪能力信息存储在数据库中，有些工具可以把每个需求与外部文件相联系（如微软的Word文件、Excel文件、图形文件等），以补充需求说明。

- 以文档为核心的工具：

使用Word或Adobe公司的FrameMaker等字处理程序制作和存储文档。

需求管理工具

工具	公司	类型
Caliber-RM	Technology Builders, Inc	以数据库为核心
DOORS	Quality Systems and Software, Inc	以数据库为核心
QSSrequireit	Quality Systems and Software, Inc	以文档为核心
RequisitePro	Rational Software Corporation	以文档为核心
RTM Workshop	Integrated Chipware, Inc	以数据库为核心
Vital Link	Compliance Automation, Inc	以文档为核心

需求分析

3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体-联系图

3.5 数据规范化

3.6 状态转换图

3.7 其他图形工具

3.8 验证软件需求

3.9 小结