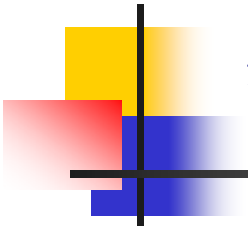
二、需求获取技术

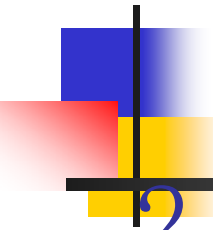
需求开发技术

- 建立负责需求的团队
- 需求调研(获取)
  - 问题分析
  - 需求启发、导出
- 需求分析
  - 用例分析技术
    - 有效的用例实践&如何编写有效的用例
    - 有效的用例模式
  - 结构化需求分析



需求开发要解决的二个问题

- 如何获取正确的需求？
- 如何表达需求？



2、1 建立负责需求的联合团队

2.1建立并利用负责需求的联合团队

- 团队组成
  - 具有编程背景的人,以获得描述所要求的准确性和
  - 熟悉系统业务规则的人
  - 了解系统如何应用的人（熟知在实际中如何使用系统的人）
  - 具有决策权的人（能够决定放弃和保留）
- 人员素质
  - 知识丰富
  - 尽职尽责，坚定不移，态度积极
  - 出色的沟通能力
  - 尊生他人的观点
  - 丰富的判断力和一个实用可行的方法

2.1 建立并利用负责需求的联合团队

- 团队职责
  - 就系统目标综合描述达成一致
  - 需求调研、导出需求
  - 分析需求
  - 检查客户提供的需求[illegible]选出：
    - 不良的需求、不真实的[illegible]
  - 修改或删除不良或不真[illegible]需求
  - 确定需求优先级
  - 管理需求变更

2.1建立并利用负责需求的联合团队

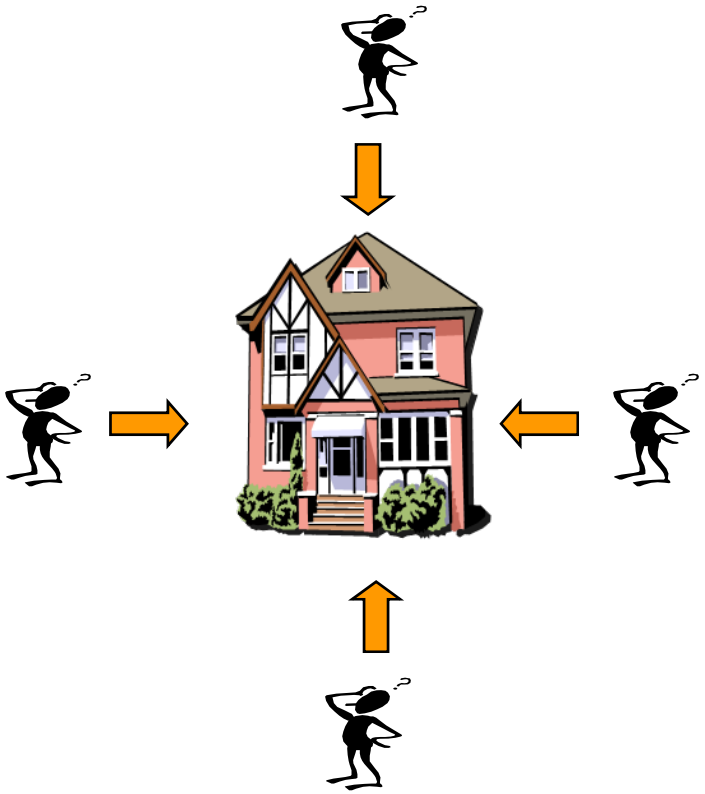
- 团队聚会
  - 经常性
  - 到少每周一次

成员素质要求

- 1.思维方式
- 2.写作策略

传统思维的误区

——局限性思维



Lateral Thinking

传统思维的误区

——局限性思维

局限性思维容易——

1、从自身的角度进行考虑；

2、从片面的角度进行考虑；

3、情绪会影响思维的效率。

替代传统思维的方法

——平行思维

平行思维是一个管理我们思维本身的一种方法。它将我们的思维从不同侧面和角度进行分解，分别进行考虑，而不是同时考虑很多因素。

平行思维的工具

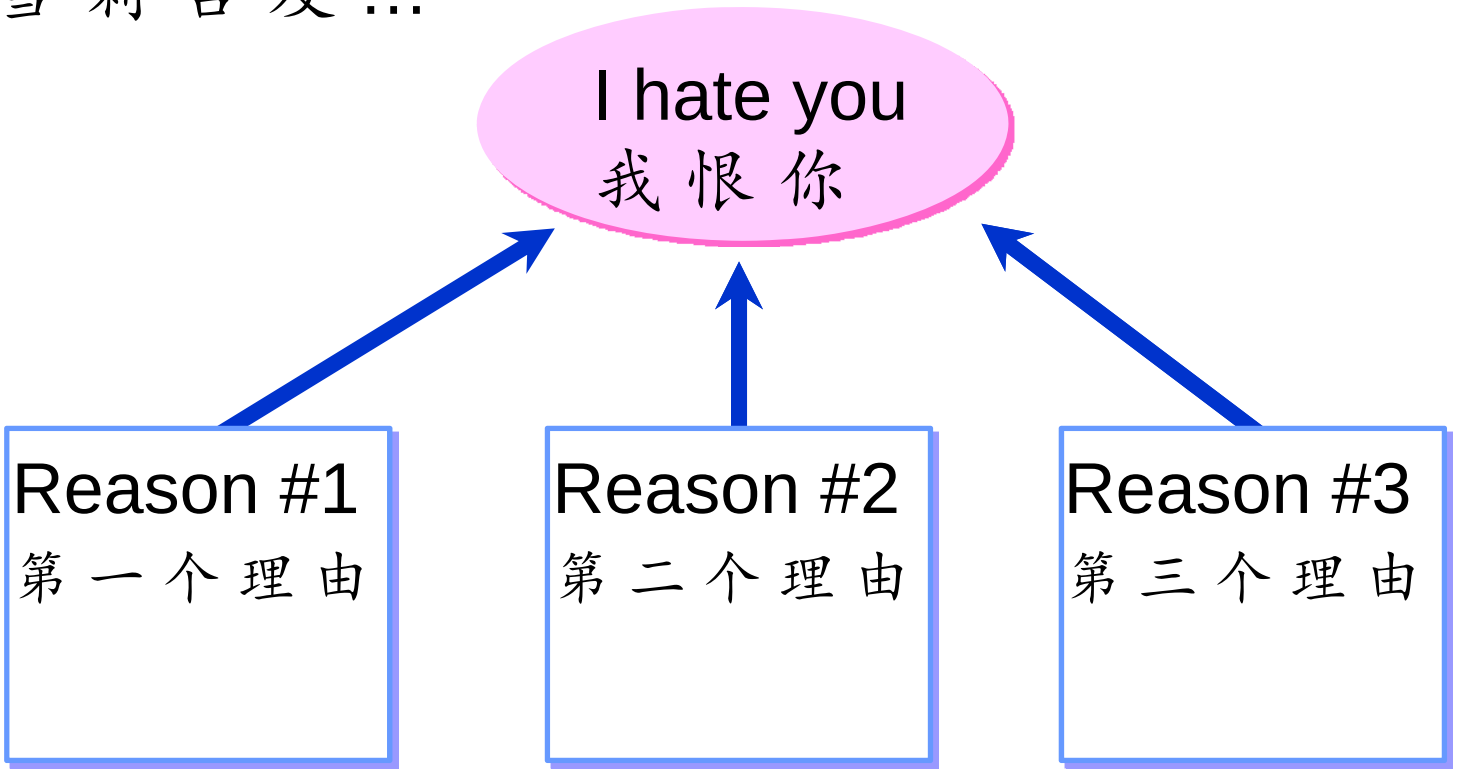
——6顶思考帽

- 一种有效地获取信息的技巧 （白帽）
- 决策时恰当的情感、直觉和预感（红帽）
- 如何在评估中使用逻辑性正面探索方式（黄帽）
- 在评估中恰当、有限地使用谨慎地逻辑性反面探索方式（黑帽）
- 创造性地真正意义及设计创造性解决问题的基本工具（绿帽）
- 变换认知——一种有效地主持会议不可或缺地技能（蓝帽）

金字塔式写作

WRITING BACKWARDS (PYRAMID PRINCIPLE)
从结论说起(金字塔写作原则)

- Dear Shirley . . .
  雪莉吾友...



SOLVING THE PROBLEM
解 决 问 题

Facts
事 实

Analyses
分 析

Conclusions
结 论

Recommen-dations
建 议 方 案

COMMUNICATING THE SOLUTION
沟 通 解 决 问 题 的 办 法

Recommendations
建 议 方 案

Actions/steps/reasons
行 动 / 步 骤 / 理 由

Facts/ analyses
事 实 / 分 析

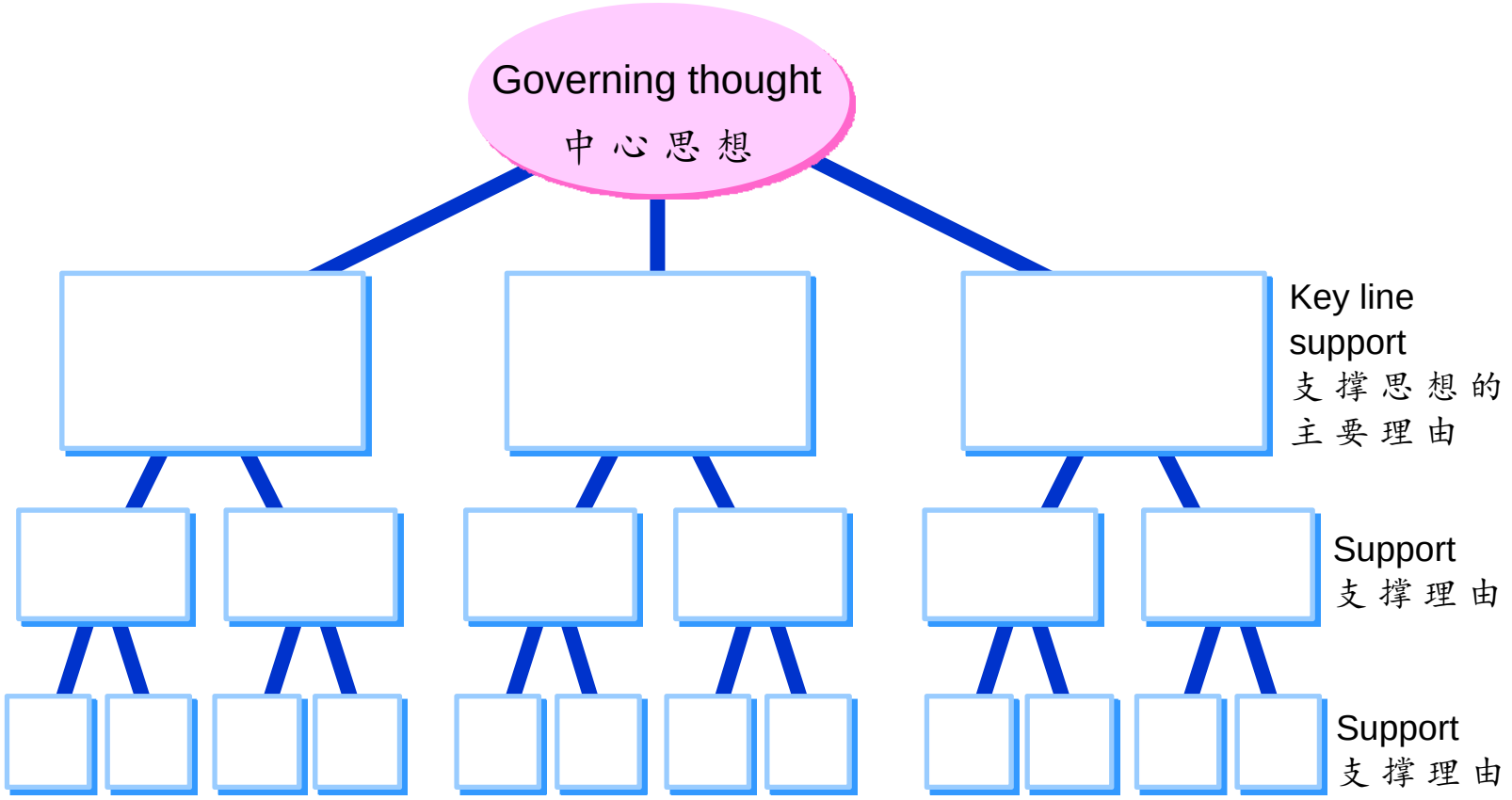
What should be done?
该 做 什 麽 ?

How or why should it be done?
为 什 麽 该 做 这 些 ?
如 何 做 ?

How do you know the recommendations are sound?
怎 麽 知 道 这 建 议 方 案 是 完 善 的 ?

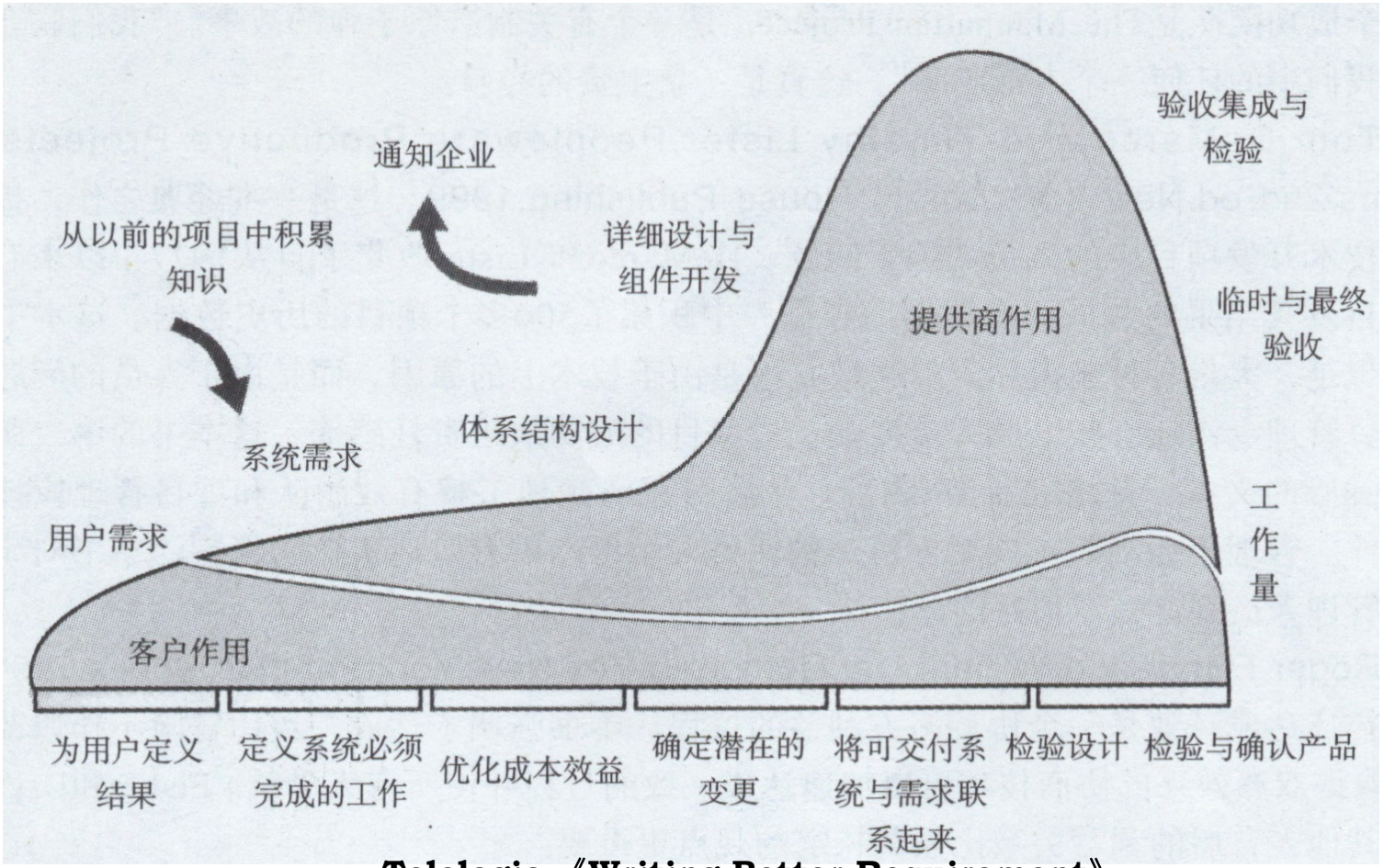
DEVELOPING A STRUCTURE FOR "WRITING BACKWARDS"

如何建构「从结论说起」的作品



2.建立并利用负责需求的联合团队

- 强调客户和供应商的作用



Telelogic 《Writing Better Requirement》

2.建立并利用负责需求的联合团队

强调客户和供应商的作用

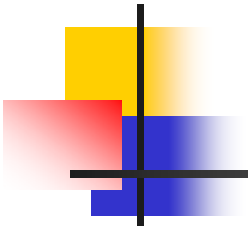
- 要求客户：
  - 建立项目领导小组
  - 指定一名客户代表
  - 指定一名需求确认人
  - 指定一种需求沟通机制
  - ……

建议的需求开发过程

- 1. 定义项目的前景和范围
- 2. 确定用户类
- 3. 在每个用户类中确定适当的代表
- 4. 确定需求决策者和他们的决策过程
- 5. 选择你所用的需求获取技术
- 6. 运用需求获取技术对作为系统一部分的用例进行开发并设置优先级
- 7. 从用户那里收集质量属性的信息和其它非功能需求

建议的需求开发过程

- 8. 详细拟订用例使其融合到必要的功能需求中
- 9. 评审用例的描述和功能需求
- 10. 如果有必要，就要开发分析模型用以澄清需求获取的参与者对需求的理解
- 11. 开发并评估用户界面原型以助想像还未理解的需求
- 12. 从用例中开发出概念测试用例
- 13. 用测试用例来论证用例、功能需求、分析模型和原型
- 14. 在继续进行设计和构造系统每一部分之前，重复6 ~ 1 3步



需求获取的过程

- 定义需求开发过程
- 定义项目愿景与范围
- 将用户群分类并归纳各自特点
- 选择产品代表
- 建立核心队伍
- 确定用例
- 确定系统事件和响应
- 举行进一步需求获取的讨论
- 观察用户如何工作
- 检查问题报告
- 重用需求

确定产品愿景与范围

- 业务需求位于需求链中的最顶层，定义了软件系统的愿景与范围。用户需求和软件功能需求都必须符合业务需求设定的前景和目标。无助于系统达到业务目标的需求都不应包含在SRS中
- 产品愿景将所有涉众统一到一个方向上。愿景描述了产品用来干什么，它最终会是什么样子
- 项目范围确定当前项目要解决产品长远规划中的哪一部分。范围描述了产品应包括的部分和不应包括的部分。范围的说明在包括与不包括之间划清了界线。也就是说，范围同时定义了项目的限制。项目范围的细节体现于开发团队为项目定义的需求基线
- 愿景关系到整个产品。当产品战略定位或者信息系统的业务目标随时间发生改变时，愿景也会随之变化，但这种变化相对缓慢
- 范围则只与一个特定的项目或实现产品功能的下一增量某次迭代相关。与愿景相比，范围更为动态，因为项目经理可以在项目进度、预算、资源和质量的约束内对每个版本的内容进行调整

项目愿景和范围的文档

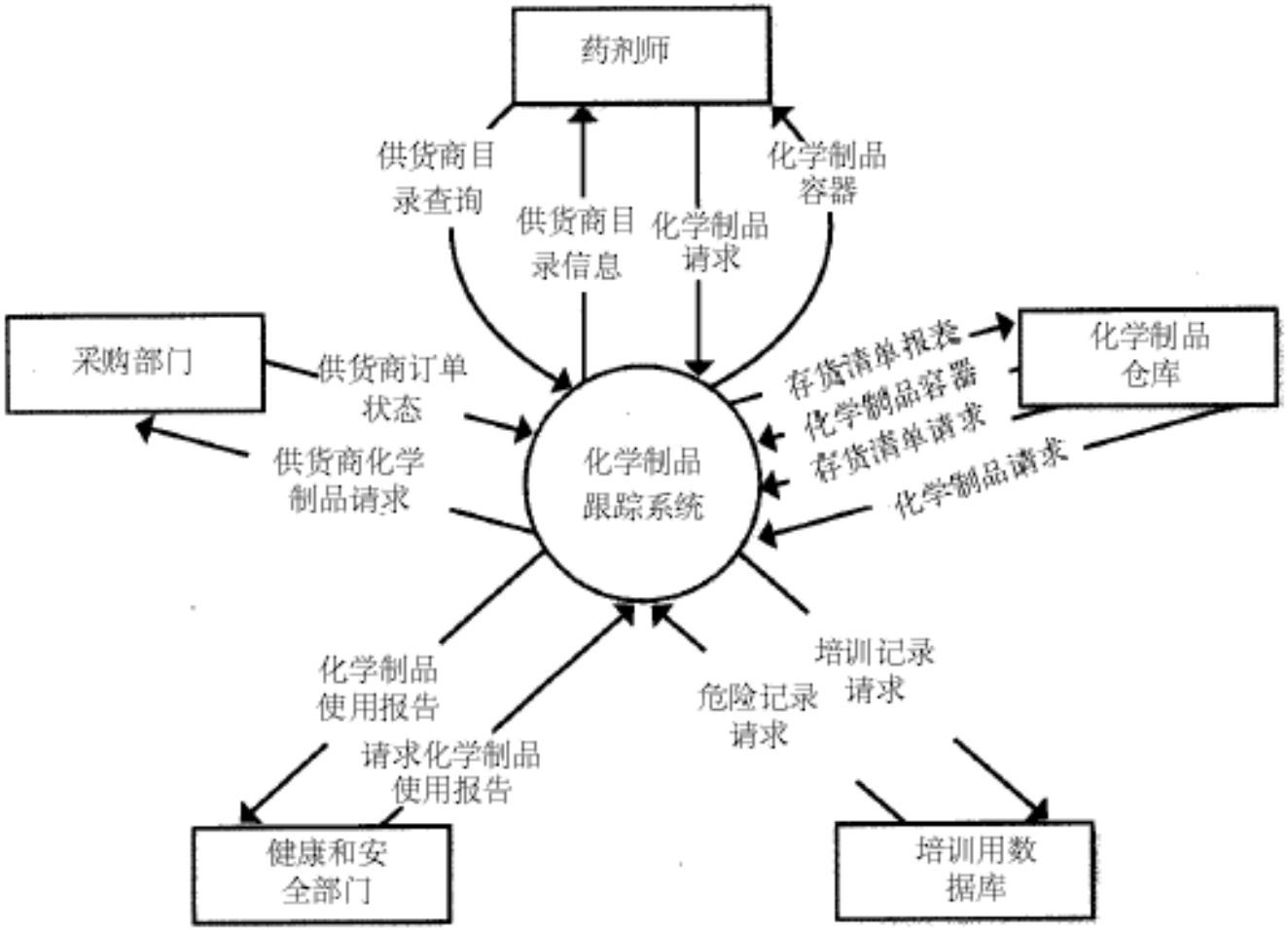
- a. 业务需求
  - a.1 背景
  - a.2 业务机会
  - a.3 业务目标与成功标准
  - a.4 客户与市场需求
  - a.5 业务风险
- b. 解决方案的愿景
  - b.1 愿景陈述
  - b.2 主要特征
  - b.3 假设和依赖
- c. 范围和限制
  - c.1 第一个版本的范围
  - c.2 各后续版本的范围
  - c.3 限制和专用性
- d. 业务背景
  - d.1 涉众简介
  - d.2 项目优先级
  - d.3 操作环境

关联图

- 对范围的描述确立了正在开发的系统与周围所有事物之间的界线和联系
- 关联图（context diagram）用图形方式说明了这一界线。图中标出了处于系统之外并与系统有某种接口的端点（terminator），以及端点与系统间的数据流和物流（material flow）
- 按结构化分析原理，并联图是数据流图中最顶层的抽象
- 关联图可以放在项目愿景与范围文档中，或者在SRS正文或附录中，也可以作为系统的数据流模型的一部分

示例

- 右图是化学制品跟踪管理系统的关联图的一部分。整个系统在图中表示为一个圆，关联图隐藏系统的内部对象过程和数据
- 图中的箭头代表系统与端点之间传达室递的数据流（“化学制品请求”）或物品（“化学制品仓库”）
- 以矩形表示的端点则分别表示了用户类（“药剂师”）组织（“采购部门”）或硬件设备（“培训用数据库”）



需求的来源

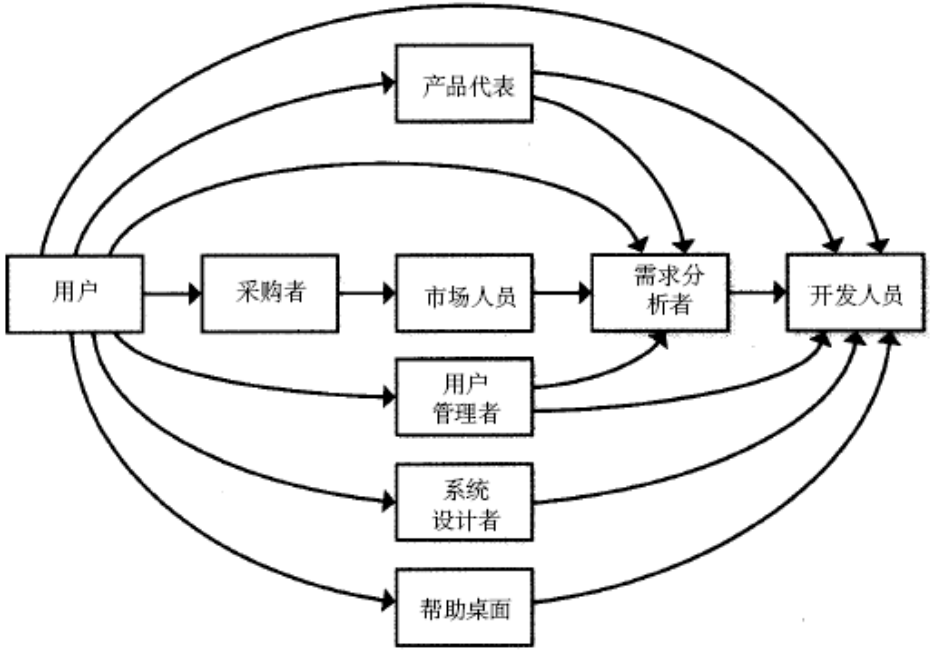
- 与潜在用户进行交谈和讨论
- 描述现有产品或竞争产品的文档
- 系统需求规格说明
- 现有系统的问题报告和改进要求
- 市场调查和用户问卷调查
- 观察用户如何工作
- 用户工作的场景分析
- 事件和响应
  - 列出系统必须响应的外部事件和正确的响应。对于实时系统来讲特别有效，因为实时系统必须从外部的硬件设备读取数据流、错误代码、控制信息和中断并对它们进行处理

用户类

- 产品的不同用户之间在很多方面存在差异。比如，使用产品的频率、用户在应用领域的经验和使用计算机系统的技能、所用到的产品功能、为支持业务过程所进行的工作、访问权限和安全级别等
- 可以根据这些差异将用户分为若干不同的用户类（user class）。关联图中系统外的端点就可能是用户类。用户类是产品用户的子集，产品用户则是产品客户的子集，而产品客户又是产品涉众的子集
- “化学制品跟踪管理系统”的用户类：
  - 药剂师
  - 采购者
  - 化学制品仓库人员
  - 卫生和安全人员

用户代表

每种类型的项目（包括企业信息系统、商业应用软件、集成系统、嵌入式系统、Internet应用程序等）在获取需求时，都需要挑选合适的用户代表来提供各类用户的功能需求和非功能需求。用户代表应当自始至终参与项目的整个过程，而不是仅参与最初的需求阶段。每一类用户都应该有自己的代表



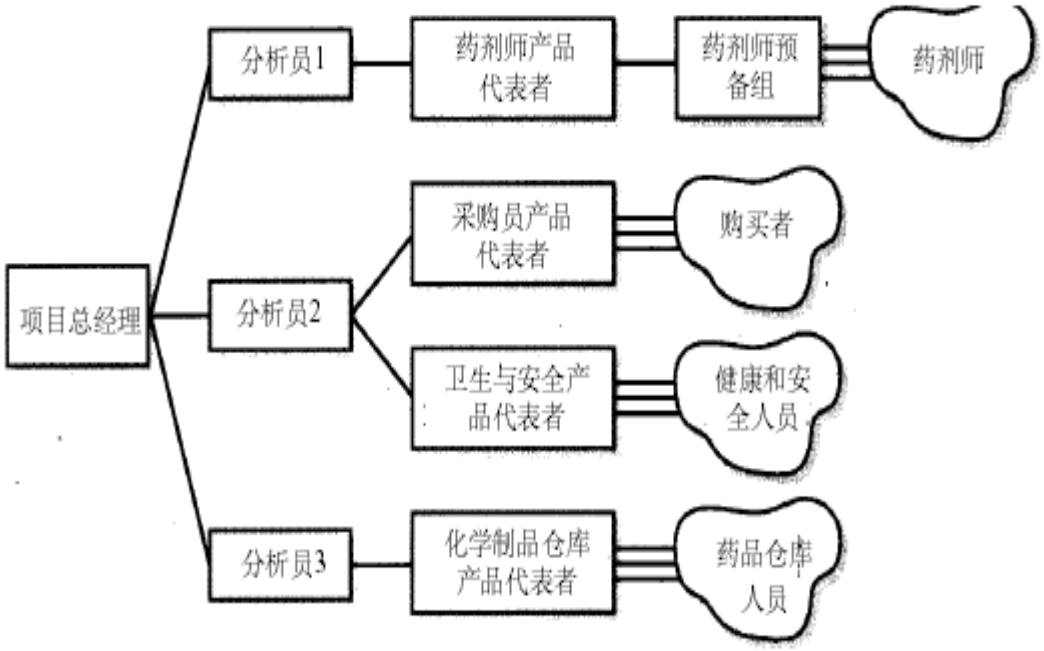
对用户代表的要求

- 为了帮助用户代表取得成功，必须把对用户代表的工作要求写成文档。下表列出了用户代表要进行的一些工作（不是每个用户代表都需要做所有这些工作）

类别	工作
计划	推敲产品的范围与限制、定义与其他系统的接口、评估新系统对业务操作的影响、宣义从现有系统到新系统的过渡方案
需求	收集其他用户的需求、开发使用范例和用例、解决用户提出的需求中的冲突、确定实现的优先级、确定质量和性能需求、评估用户界面的原型
确认和验证	评审需求文档、定义用户验收系统的标准、根据用例开发测试用例、提供测试数据集、执行beta测试
用户帮助	编写用户手册和帮助文档、准备培训资料、向同行演不产品
变更控制	评估缺陷修改请求，确定其优先级、评估改进请求，确定其优先级、评估需求变更对用户和业务流程的影响、参与变更决策

设置多位用户代表

- “化学制品跟踪管理系统”有四个主要的用户类，此需要从内部用户中选多个用户代表。右图说了项目经理如何组织需分析员和用户代表小组
- 用户代表并不是全职，他们每个星期都要花数小时为项目工作
- 三个分析员与四个用户表一起协作进行需求获分析，并把它们的需求写成文档（采购人员和生与安全用户类很小，们的需求也不多，因为一位需求分析员负责与两类的用户代表合作）
- 最后，由一名分析员综合这些信息，并编写到软件需求规格说明中



谁做出决策

- 当不同用户类提出的需求发生矛盾时，必须有人来协调、解决这些矛盾，并做出决策
- 在每个项目的早期阶段，必须决定谁是需求问题的决策者
- 不过，由谁来做出决策并没有一个普遍适用的答案
- 下面列出了项目中可能发生的一些需求矛盾并推荐了一些处理办法：
  - 如果是个别用户之间的分歧，则由用户代表来裁决
  - 如果不同的用户类或客户群提出的需求发生冲突，应支持最重要的用户类或对产品的商业前景影响最大的客户群提出的需求
  - 不同的企业客户都可能要求按他们自己的喜好设计产品，解决办法还是依据项目的业务目标来确定哪些客户对项目的成败影响最大
  - 有时客户经理所提出的需求与其所在部门的真正用户提出的需求相冲突。尽管用户需求必须与业务需求一致，但不属于用户类的经理应该服从于用户代表
  - 当开发人员对产品的想法与客户的要求不一致时，通常应由客户做出决定
  - 类似地，营业员销和生产管理部门提出的需求与开发人员对产品的想法发生冲突时，营业员销部门的意见应受到更多的重视

需求调研的中间工作成果

- 业务领域的当前业务说明
- 业务领域的当前问题清单
- 业务目标、关键问题的清单
- 未来系统的大型结构的构思
- 切实的可能性
- 后果及风险
- 相关人员的认可
- 相关人员间的冲突决议
- 最终需求
- 需求的优先级
- 完整性

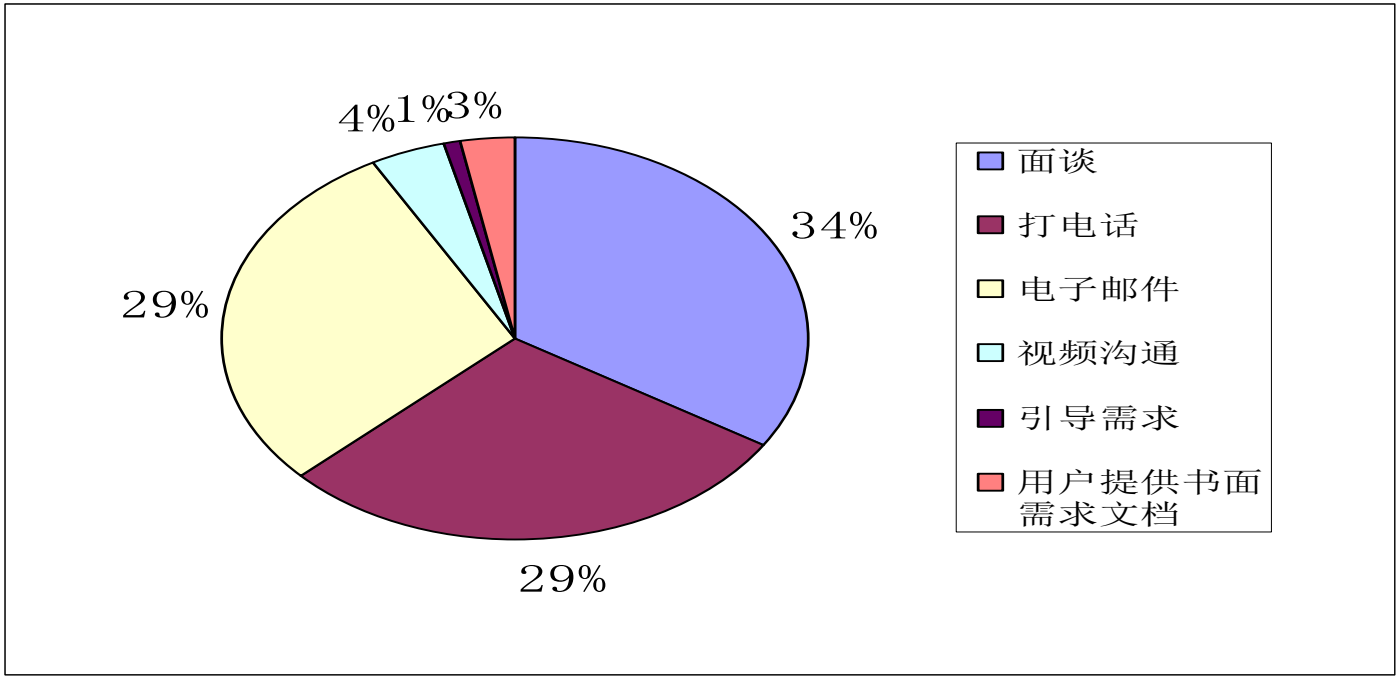
用户参与

- 作为设计组、专题讨论会的成员，参与设计用户界面
- 作为知识来源，提供任务、商业过程的业务情况
- 参与讨论会，提出构想、确定问题
- 作为测试用户，在验收时测试系统，检查它能否正常工作
- 作为审查者评估最终需求和用户界面
- 进行可用性测试
- 作为项目管理委员会的成员

导出需求的常用技术

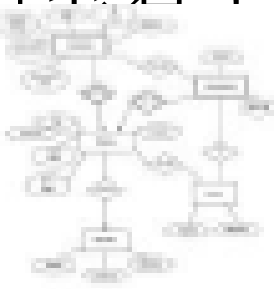
- 面谈
- 观察
- 示范
- 文档研究
- 问卷调查
- 原型开发
- 需求专题研讨会
- 研究类似组织
- 自由讨论
- 情节串联板
- 应用用例
- 角色扮演

国内的主要需求调研和交流方式



一、文档研究

- 工作中的表格、文件、便函…
- 工作报告…
- 规章制度、业务规程、程序文件、质量手册
- 旧系统的设计文档（如果有）
- 旧系统
- ……
- 收集和分析现有文档



文档研究

- 要点
  - 一般作为访谈之前的准备
  - 有价值内容的比例少，随时做笔记
  - 有助于得到：组织结构图、业务对象模型
    - 岗位责任分工、标准业务流程……

二、调查问卷

- 用调查问卷收集事实
- 调查问卷的类型
  - 自由格式
  - 固定格式
    - 多选
    - 分类
    - 队列
- 开发调查问卷

调查问卷

- 优点
  - 回答快速、方便
  - 直观的结果
  - 可以匿名回答
  - 列表式的响应、分析快速
- 缺点
  - 相关的问题不能事先决定
  - 响应的数量少
  - 不能保证所有问题被回答
  - 问题可能被误解
  - 难以继续用户的模糊回应
  - 不能发现行为语言
  - 好的调查问卷很难去准备
  - 难以探索新的领域

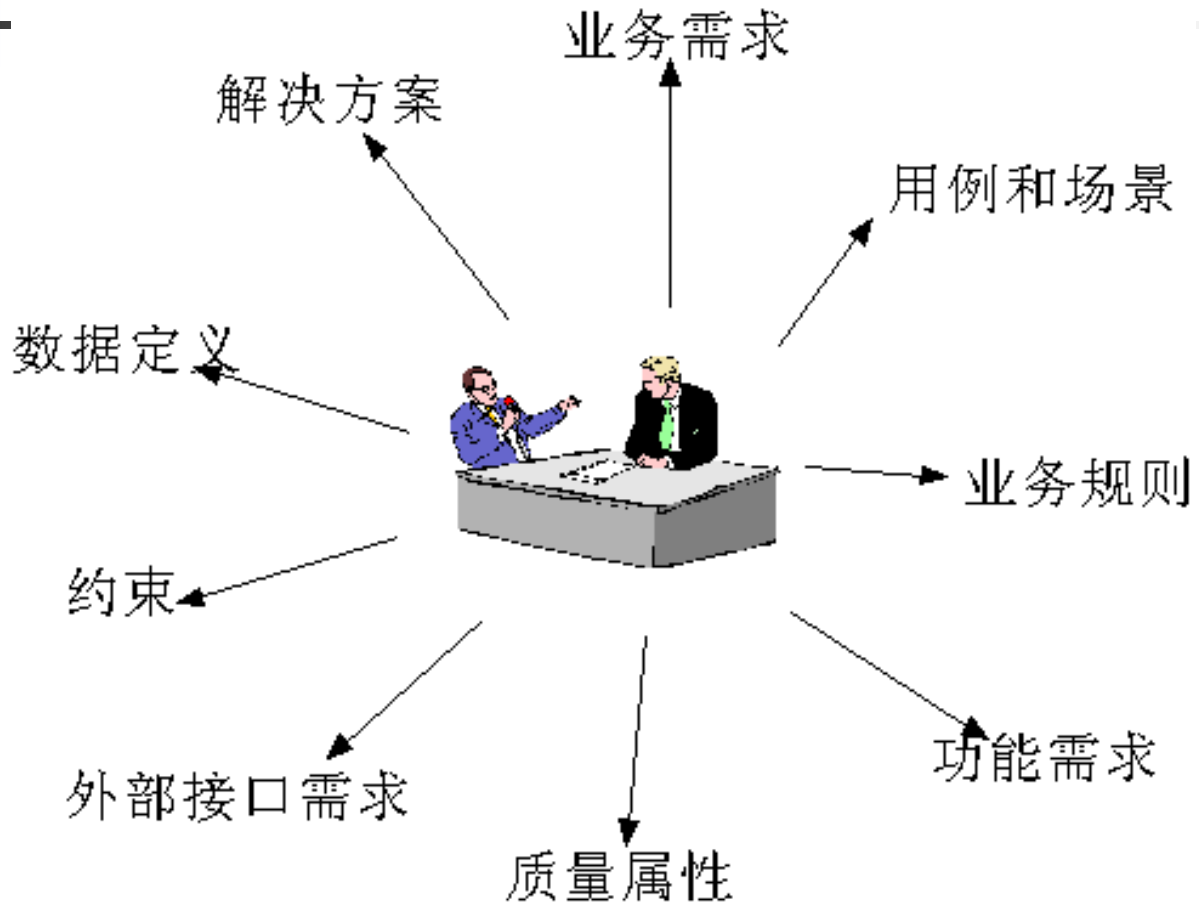
三、面谈

- 用户面谈是理解商业功能和商业规则最有效的方法
- 面谈之前
  - 确立面谈的目的
  - 确定要包括的相关用户
  - 确定参加会议的项目小组成员
  - 建立要讨论的问题和要点列表
  - 复查有关的文档和资料
  - 确立时间和地点
  - 通知所有的参加者有关会议的目的、时间和地点

面谈

- 进行面谈
  - 衣者得体，准时到达
  - 寻找异常和错误情况
  - 深入调查细节
  - 详细记录
  - 指出和记录下未回答和未解决的问题
- 面谈之后
  - 复查笔记的准确性、完整性和可理解性；
  - 把所收集的信息转化为适当的模型和文档；
  - 确定需要进一步澄清的问题域
  - 适当的时候向参加会议的每一个人发一封感谢信

用户表达内容的层次



四、观察（做学徒）

亲身体验用户的工作—最直接的工作

观察（做学徒）

- 师傅
  - 经验丰富的师傅（系统就是为了分担他的经验）
  - “测不准”原理

观察（做学徒）

……学徒

- 诚心求教，专心体会
- 问题
  - 下一项目工作是什么
  - 这样做的原因是什么
  - 这种情况出现得频繁吗
- 重点关心
  - 完成哪些工作
  - 完成一项工作的所需时间
  - 操作次数
  - 出现的错误和混乱

观察（做学徒）

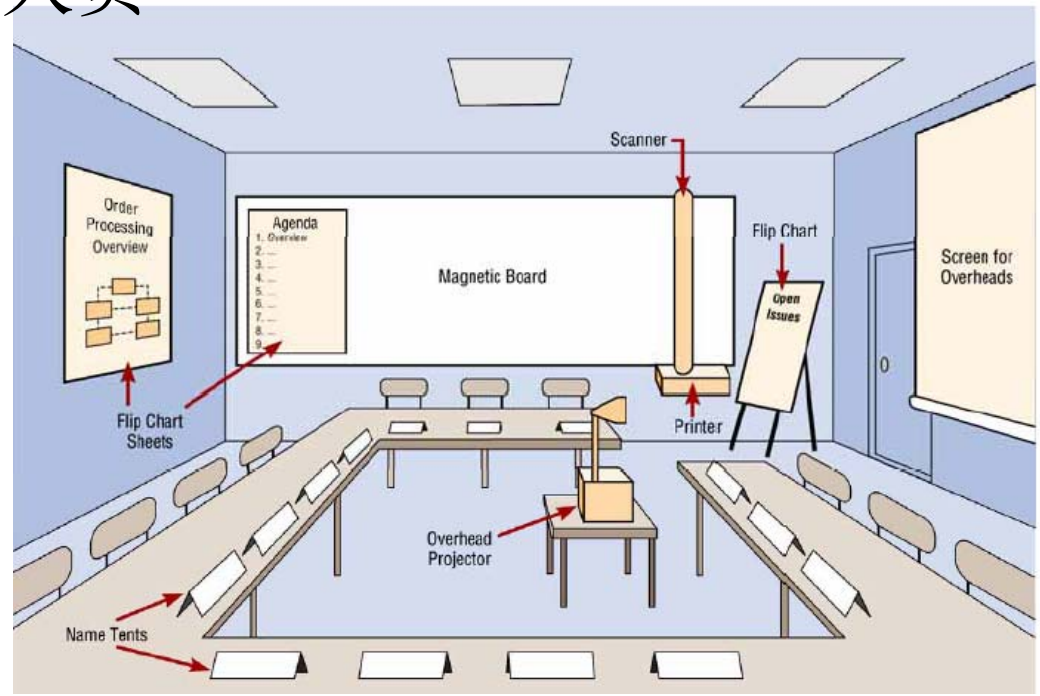
- 其他
  - 效率较低
  - 有遗漏

五、需求专题讨论会

- 通过让所有的相关人员一起参加某个单一的会议来定义需求或设计系统，也称联合应用设计会议（JAD）
- 干系人员在短暂而又紧凑的时间段内集中在一起，一般为1~2天，与会者可以在应用需求上达成共识、对操作过程尽快取得同意的意见；
- 协助建立一支高效团队，围绕一个目的：项目的成功；
- 所有的人员都畅所欲言；
- 促进用户和开发团队之间达成共识；
- 能够接露和解决那些妨碍项目成功的行政问题；
- 最终很快产生初步的系统定义

需求的专题讨论会

- 专题讨论准备
  - 通知与确认与会人员
  - 安排日程
  - 安排场所



需求专题讨论会

- 举行会议
  - 可能出现行政间的责备和冲突，应掌握讨论气氛并控制会场；
  - 最重要的部分是自由讨论阶段，这种技术非常符合专题讨论会的气氛，并且营造出一种创造性、积极的氛围，同时还可以获取尽可能多的意见；
  - 分配会议时间，记录所有的言论。
- 会后
  - 形成会议纪要和遗留问题列表
  - 第一时间确认
  - 协商进一步需求获取方式、时间和对象

六、观察业务流程和操作

- 掌握用户如何实际使用一个系统以及到底用户需要哪些信息，最好的办法是亲自观察用户是如何完成实际工作的
  - 对办公室进行快速浏览，了解布局、设备要求和使用、工作流总体情况；
  - 安排几个小时观察用户是如何实际完成他们的工作，理解用户实际使用计算机系统和处理事务的细节；
  - 向用户一样接受训练和做实际工作，发现关键问题和瓶颈；
- 注意：观察可能使用户紧张

遗漏的需求

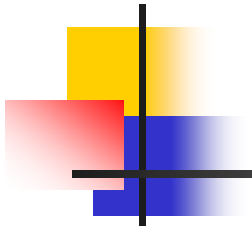
- 遗漏需求是最常见的需求缺陷类型。下面的方法用于帮助找出以前未曾发现的需求：
  - 将高层需求分解得足够细，让用户的真正需求显露出来
  - 务必让所有的用户类都提出他们的意见。确保每个用例至少有一个确定的执行者（actor）
  - 跟踪系统需求、用例、事件-响应表以及业务规则，直至详细的功能性需求，确保需求分析员推导出了所有必需的功能
  - 检查边界值，查找被遗漏的需求
  - 用多种方法表达需求信息

CRUD矩阵

- 查找遗漏的需求的精确方法是CRUD矩阵。CRUD代表创建（create）、读取（read）、修改（update）、删除（delete）
- CRUD矩阵将系统行为与数据实体（单个数据项或数据项集）联系起来，确保了解每个数据项在何处及如何被创建、读取、修改和删除
- 下页的表给出了一个CRUDL矩阵示例，该示例是“化学制品跟踪管理系统”的一部分。每个单元格说明最左例中的用例如何使用其他列所代表的数据实体。用例可以创建、读取、修改、删除和列出（L）数据实体。生成CRUDL矩阵后，可以检查这个5个字母中是否有一个在同一列的所有单元格都没有出现。比如，如果某个业务对象被修改却从未被创建，它从何而来？在表中，申领人（下订单购买某种化学制品的人）列中所有单元格都不包含D，也就是说，任何用例都不能将申领人从化学制品订购者的名单中删除。对此有三种解释：
  - 删除申领人并不是化学制品跟踪管理系统应提供的功能
  - 遗漏了删除申领人的用例
  - “编辑申领人”用例不正确，该用户应该允许用户删除申领人，但现在却遗漏了这一功能

	订单	化学制品	申领人	厂商目标
设置订单	C	R	R	R，L
改正订单	U，D		R	R，L
管理化学制品目录		C，U，D		
订单报告	R	R，L	R，L	
编辑申领人			C，U，L	

问题？



谢 谢!