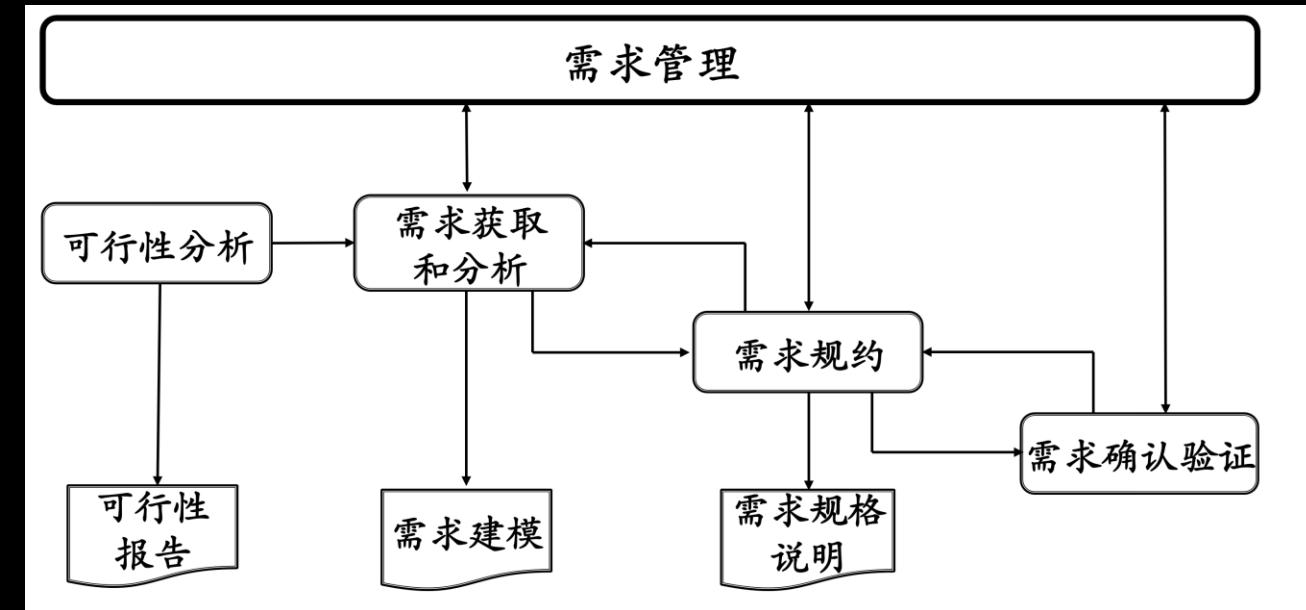


Requirements



Jianyong Wang(王建勇)

Department of Computer Science and Technology
Tsinghua University, Beijing, China

内容提要



软件需求

- 需求是对应当执行的任务的规范说明，描述系统的行为特性或属性，是一种对系统开发进程的约束。[Ian&Pete 1997]
 - 功能性需求：描述系统应该做什么，即为用户和其他系统完成的功能、提供的服务，规约了系统必须执行的功能【是需求主体，没有它就没有非功能需求】
 - 非功能需求，包括业务规则、外部接口、设计约束、质量属性（性能、可靠性、可维护性、安全性等）等



早上好！我是业务分析师，负责为贵公司开发一个新的员工信息系统。谢谢你同意作为这个项目的产品代言人。您给出的信息会对我们有莫大的帮助。现在，您能告诉我您想要什么吗？

呃，我想要什么？.....



从哪里说起呢？让我想想.....



对，新系统要比旧系统速度快！



还有，你知道吗？如果员工的姓名很长，旧系统就会崩溃，这时我们就得给客服打电话，让他们来替我们输入姓名。因此，新系统要在我们输入长姓名的时候不崩溃。



还有，现在有一个新规定，就是说我们不能再使用身份证号码作为员工的ID了，所以新系统投入使用后，我们还得更改所有ID。



哦，对了，如果我能有一份截至到本年度每位员工受过多少课时培训的报告，就太好不过了。





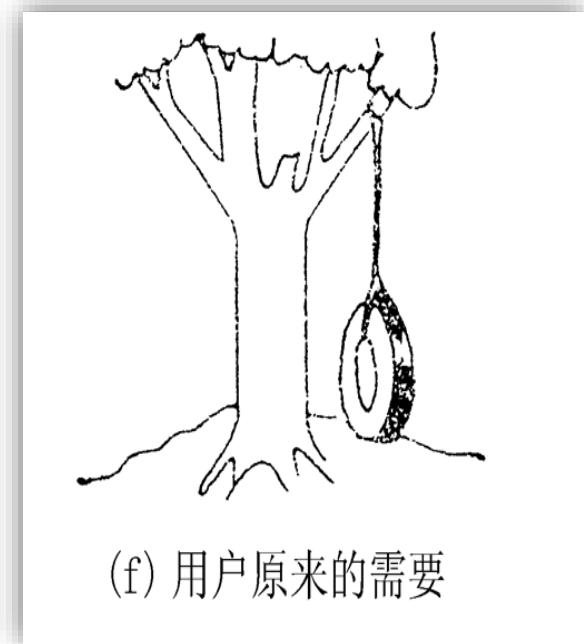
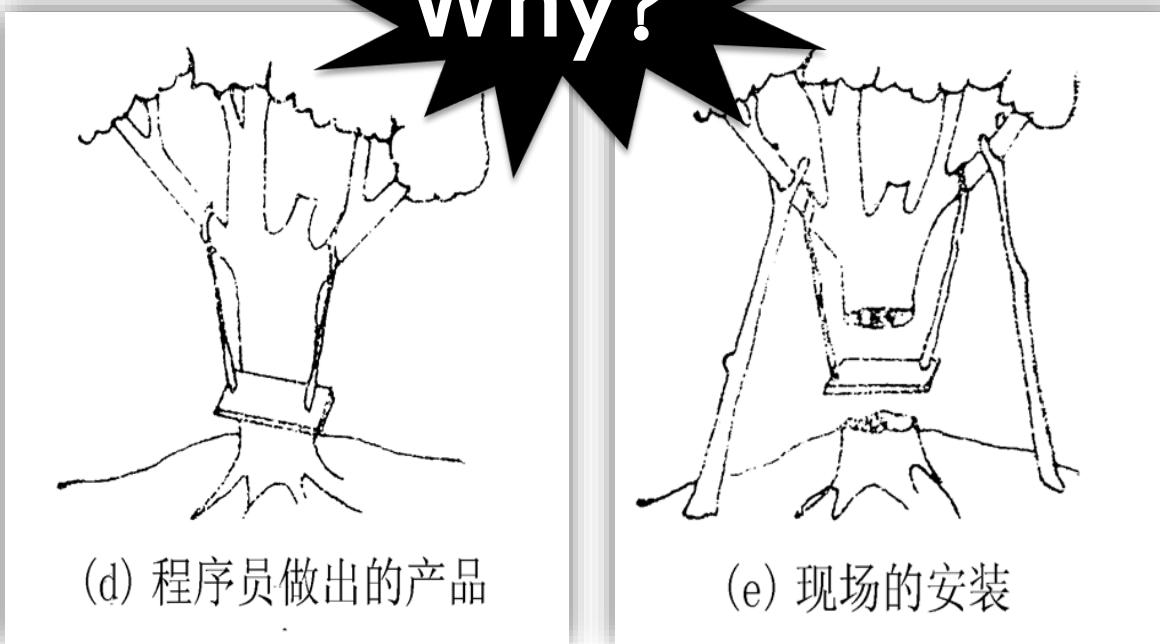
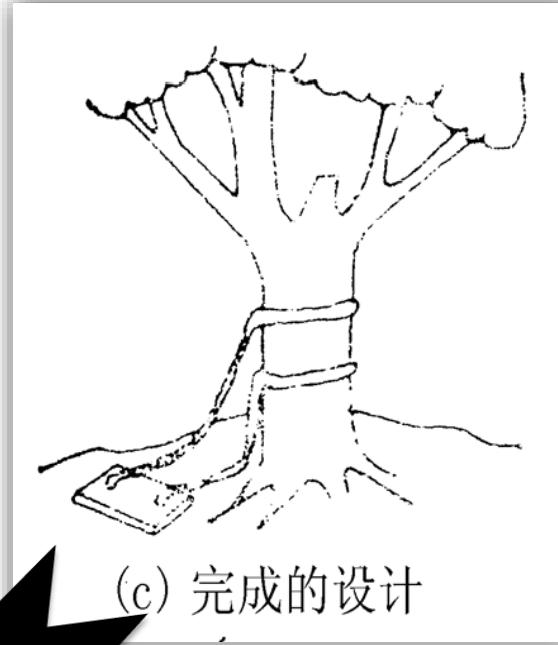
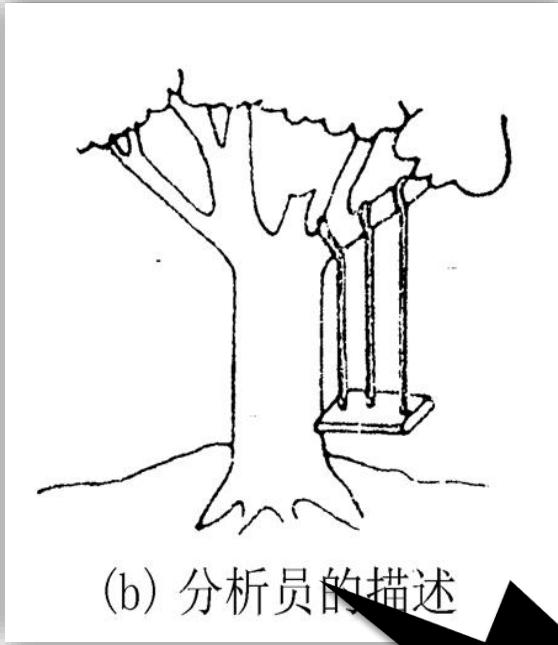
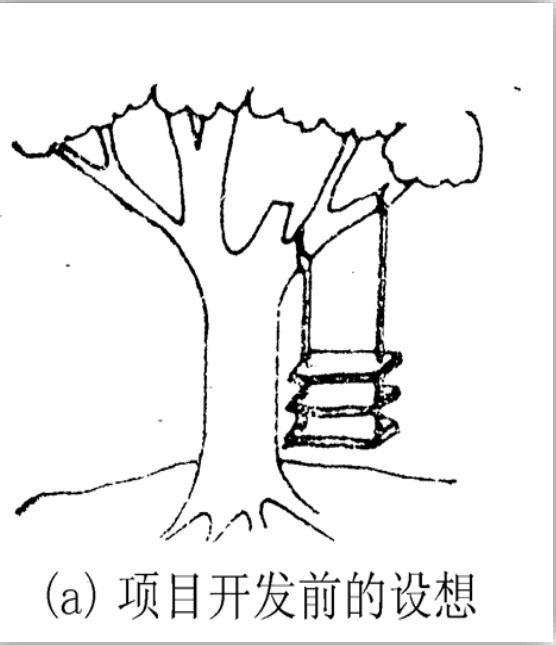
php

Python

Perl







Why?



软件开发的噩梦

- “It’s your worst nightmare. A customer walks into your office, sit down, looks you straight in the eyes, and says, “***I know you think you understand what I said, but what you don’t understand is what I said is not what I mean.***”
- WISBNWIW: What-I-Said-But-Not-What-I-Want
- Invariably, this happens late in the project, after the deadline commitments have been made, reputation are on the line, and serious money is at stake.
- All of us who have worked in the systems and software business for more than a few years have lived this nightmare, and yet, few of us have learned to make it go away.”

-- R. Young, *Effective Requirements Practices*, Addison-Wesley, 2001

Why software fails?

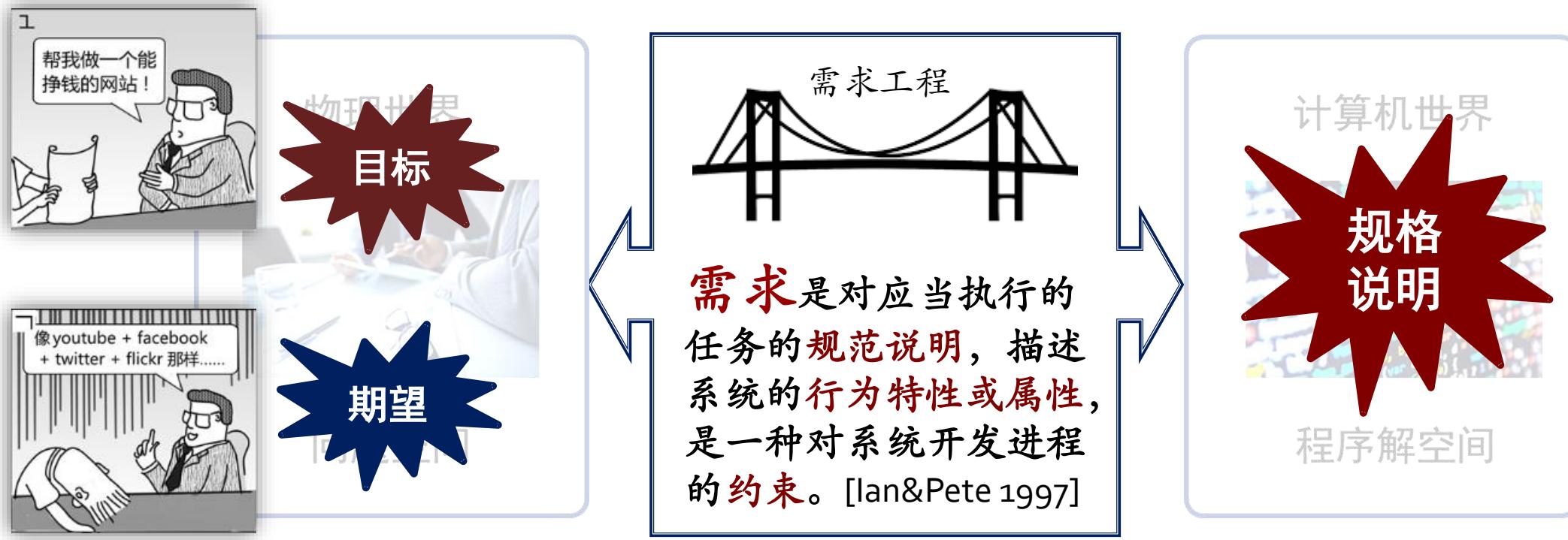
1. Incomplete Requirements	13.1%
2. Lack of User Involvement	12.4%
3. Lack of Resources	10.6%
4. Unrealistic Expectations	9.9%
5. Lack of Executive Support	9.3%
6. Changing Requirements & Specifications	8.7%
7. Lack of Planning	8.1%
8. Didn't Need It Any Longer	7.5%
9. Lack of IT Management	6.2%
10. Technology Illiteracy	4.3%
11. Others	9.9%

Source: Standish Group 1995, 350个公司, 8000个项目

“The seeds of major software disasters are usually sown in the first three months of commencing the software project.”

Capers Jones

软件需求



用户

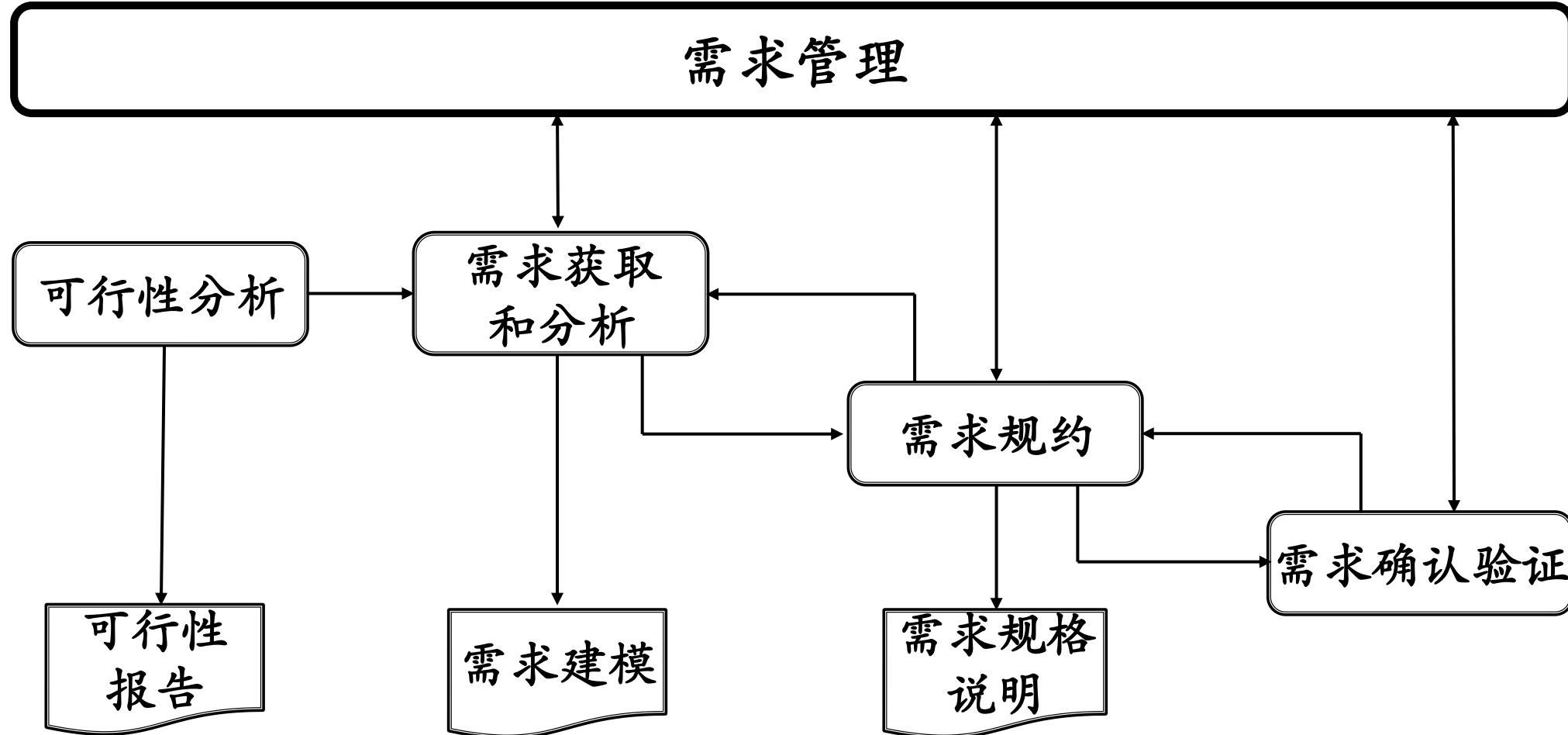


分析师



工程师

需求工程过程



内容提要

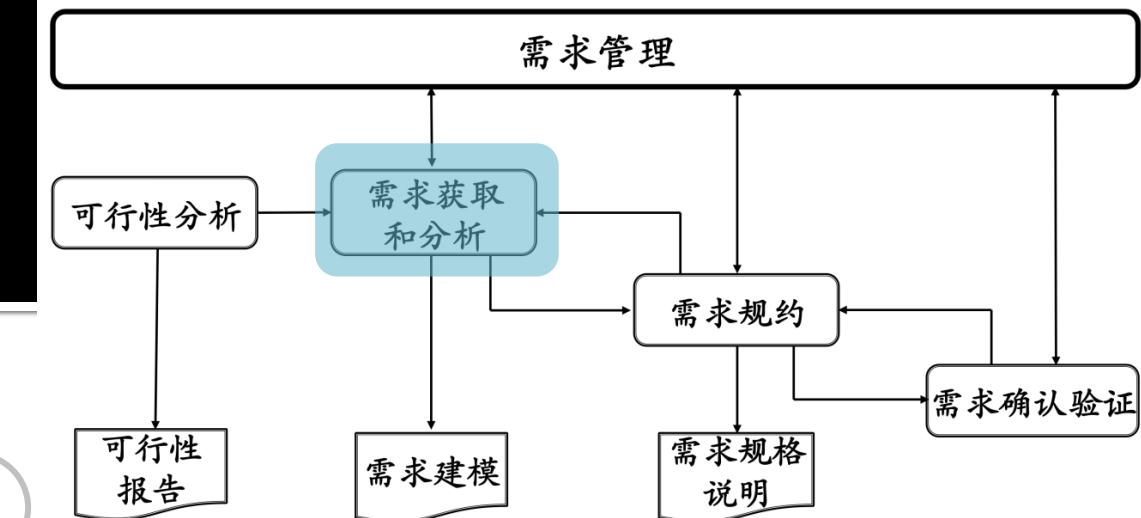
基本概念

需求获取

需求建模

需求规约

需求管理



用 户

“.....需求收集以及许多软件和系统项目工作中，普遍面临的最大挑战是人与人如何互动。尽管许多工具都可以用于帮助地理分散的人们开展有效的协作，但没有哪一种神奇的新技术能够将此自动化。.....在软件开发中，沟通总是重于计算机操作。.....”

软件项目成功因素		%	软件项目挑战因素		%	软件项目失败因素		%
1. 用户参与 User Involvement	15.9%		1. 缺少用户输入 Lack of User Input	12.8%		1. 需求不完整 Incomplete Requirements	13.1%	
2. 管理层的支持 Executive Management Support	13.9%		2. 需求及规格说明不完整 Incomplete requirements & Specifications	12.3%		2. 缺少用户参与 Lack of User Involvement	12.4%	
3. 明确需求 Clear Statement of Requirements	13.0%		3. 需求及规格说明变化 Changing requirements & Specifications	11.8%		3. 缺少资源 Lack of Resources	10.6%	
4. 合适的计划 Proper Planning	9.6%		4. 缺少管理层的支持 Lack of Executive Support	7.5%		4. 期望不切实际 Unrealistic Expectations	9.9%	
5. 合理现实的期望 Realistic Expectations	8.2%		5. 技术过时 Technology Obsolescence	7.0%		5. 缺少管理层支持 Lack of Executive Support	9.3%	
6. 小的项目里程碑 Smaller Project Milestones	7.7%		6. 缺少预算 Lack of Budget	6.4%		6. 需求及规格说明变化 Changing Req. & Spec	8.7%	
7. 称职的员工 Competent Staff	7.2%		7. 期望不切实际 Unrealistic Expectations	5.9%		7. 缺少计划 Lack of Planning	8.1%	
8. 所有权 Ownership	5.3%		8. 目标不明确 Unclear Objectives	5.3%		8. 软件不再需要 Didn't Need It Any Longer	7.5%	
9. 清晰的愿景和目标 Clear Vision & Objectives	2.9%		9. 时间安排不切实际 Unrealistic Time Frames	4.3%		9. 缺少IT管理 Lack of IT Management	6.2%	
10. 努力工作，专注的员工 Hard-Working, Focused Staff	2.4%		10. 新技术 New Technology	3.7%		10. 技术薄弱 Technology Illiteracy	4.3%	
其他	13.9%		其他	23.0%		其他	9.9%	

"The chaos report." The Standish Group (2014).

用户



微信：熟人圈

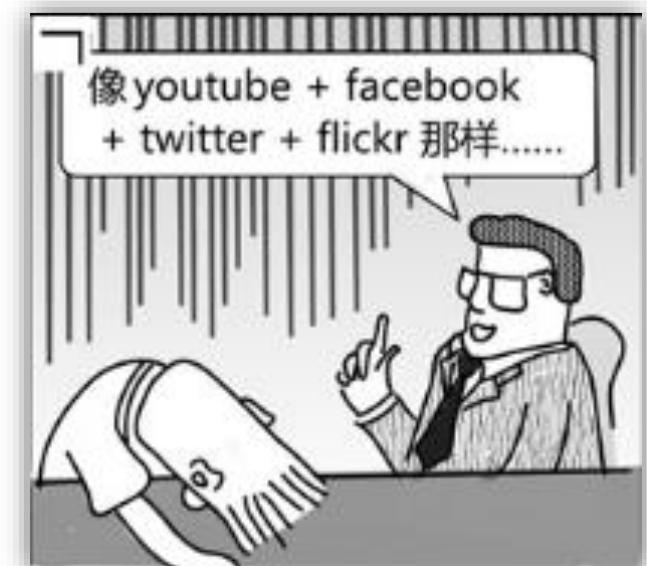


以用户需求为驱动的产品设计

理解和探求用户目标，
提取必要的系统功能。



陌陌：陌生人



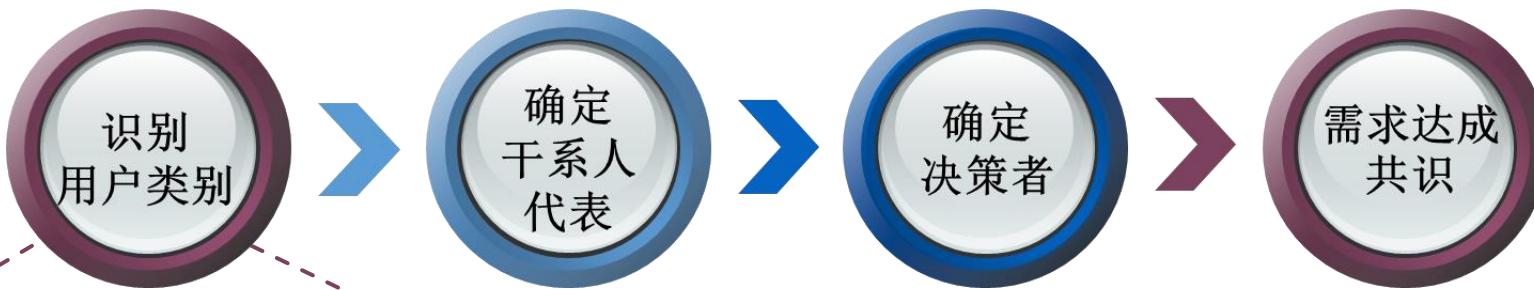
药剂师很忙.....



我们需要构建一个能够跟踪化学制剂的信息系统，这个系统应该能够跟踪我们仓库和实验室里所有的化学制剂。这样一来，药剂师就能够使用其他人剩余的化学制剂而不是总去购买新的。这会为我们节省大量经费。与此同时，健康与安全部门希望这个系统能够帮助他们大大减少向政府提供化学品使用和处理报告的工作量。你们能在五个月内及时开发出符合这些要求的系统吗？

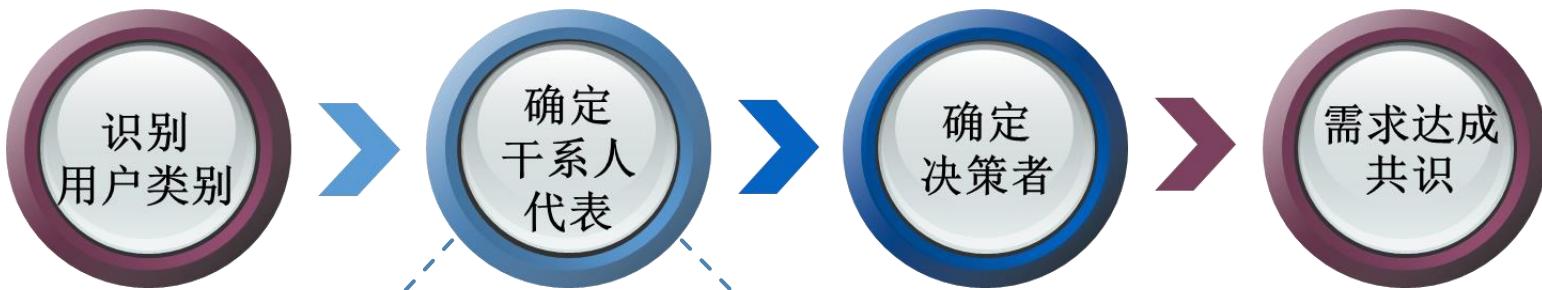
药剂师都很忙，他们没时间在你们开发之前确定所有的细节。你们的人难道就不能自己想想究竟该做什么吗？

我已经告诉了你需求了。现在就请开工吧。把你们的进度通报给我。



- 用户群的访问权限或安全级别
- 用户群在业务操作中执行的任务
- 用户群使用的特性
- 用户群使用产品的频率
- 用户群在应用领域和计算机专业技能经验
- 用户群使用的平台（台式机、笔记本、平板、智能手机等）
- 用户群的语言
- 用户群是直接还是间接与系统交互

名称	人数	描述
药剂师	大约1000人，分布在6幢办公楼中	药剂师从供应商或者化学品库房获取化学品。每个药剂师每天多次使用系统.....
采购者	5	负责处理化学品采购申请。发出并跟踪发给外部供应商的订单。
化学品库房工作人员	6个技术员，1个主管	管理员管理着超过500000个化学品容器。从3个库房中提供容器，从供应商处购买新化学品，跟踪每个入库和出库的容器.....
健康与安全部门职员	1个经理	用系统生成预定义的季度报告，报告必须符合联邦和州化学品使用和处理报告法规。.....



干系人是指积极参与项目的某个人、群体或组织，可能会受项目过程和结果的影响或影响项目的过程和结果。

- 开发组织以外
- 开发组织
- 项目组

开发组织以外

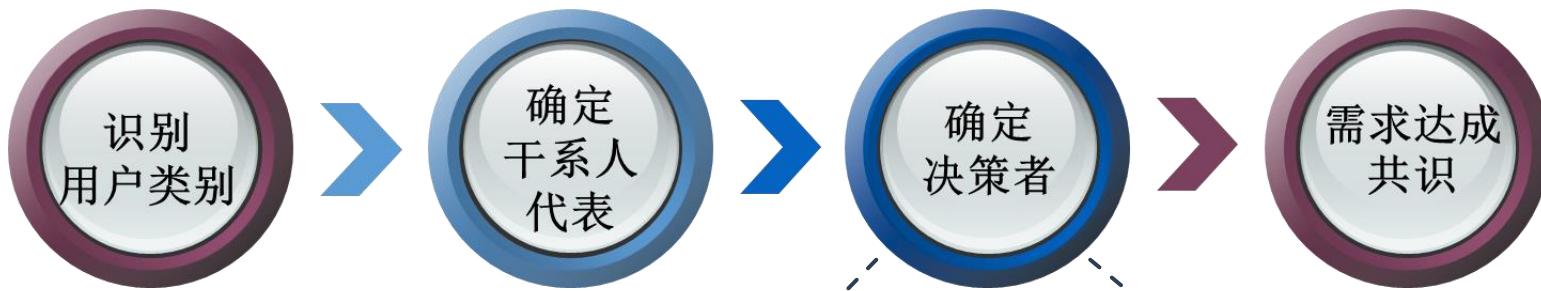
直接用户	业务管理部门	顾问
间接用户	合同签订人员	法律法规审核员
收单机构	政府机关	证明人
采购人员	领域专家	监管机构
法务人员	项目经理	软件供应商
承包商	测试版本检查人员	材料供应商
转包商	普通大众	风险投资人

开发组织

开发部门经理	销售人员	执行发起人
市场人员	安装人员	项目管理部门
运营支持人员	维护人员	生产部门
法务人员	程序经理	培训专员
信息架构师	易用性专家	投资组合架构师
公司业主	领域专家	基础架构支持人员

项目组

项目经理	测试人员
业务分析人员	产品经理
软件应用架构师	质量保证人员
设计师	文档工程师
开发人员	数据库管理员
产品负责人	硬件工程师
数据建模工程师	基础架构分析师
过程分析人员	业务方案架构师

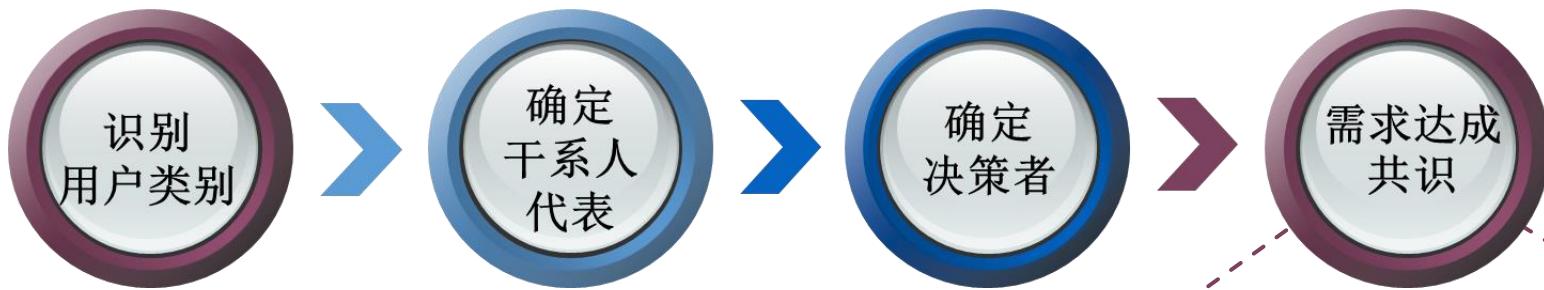


决策规则

- 决策领导做决定
 - 小组投票，少数服从多数
 - 小组投票，但结果必需获得一致通过
 - 小组讨论和协商达成共识。每个人都拥有这个决定并承诺支持。
 - 决策领导授权一个决策人
 - 小组达成决策，但允许有人有权否定小组决定
-

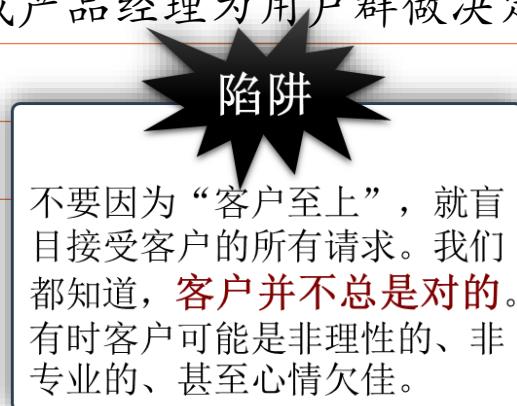
在软件项目中，需要在关键路径上做决定，或是解决一些冲突，接受（或拒绝）某个需求变更或者批准一组即将发布的需求。

决策人或决策小组对决定工作方向、保证项目进度具有重要作用。



解决需求争议的建议

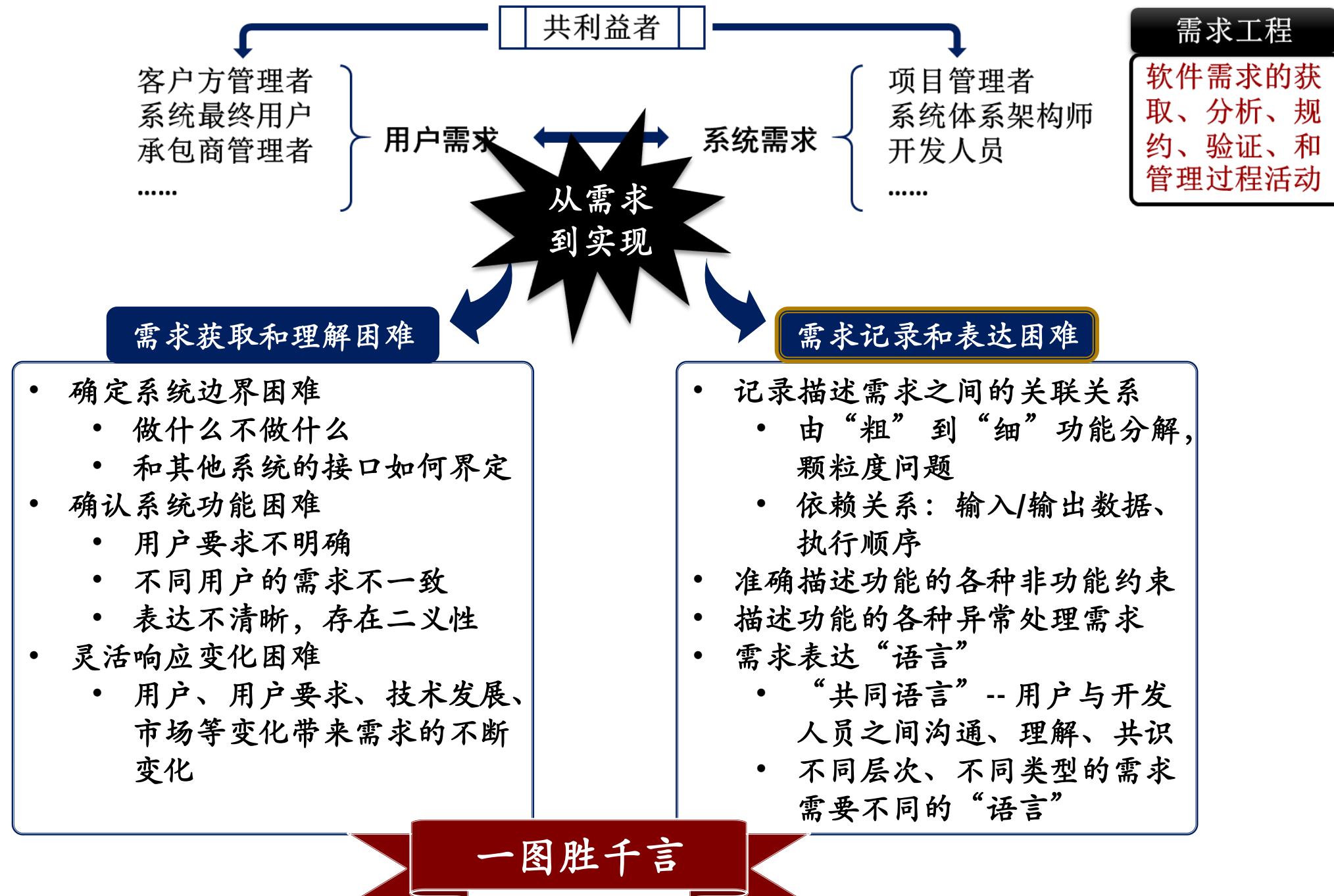
争论方	如何解决
单独用户之间	产品代言人或产品负责人决定
用户群之间	优先满足受优待的用户群
细分市场之间	对业务成功影响最大的细分市场优先
企业客户之间	由经营目标决定方向
用户和用户经理之间	产品负责人或产品经理为用户群做决定
开发人员和顾客之间	
开发人员与销售之间	



对需求或其中某部分达成一致是客户-开发人员关系的核心

- 客户确认：需求描述了他们的需要
- 开发人员确认：理解需求并且可以实现
- 测试人员确认：需求是可验证的
- 管理层确认：需求满足业务目标

需求基线，一个特定时间点的需求快照，一组需求，作为后续开发的基础。



需求样例

用户要求

MHC-PMS系统每月都要

①

产生报告来显示当月每个诊所

③

②

所开药物的成本。

④

系统需求

① 在每个月的最后一个工作日，对所开的药物、药物成本以及开药诊所应该产生一个汇总报告。

② 系统应该在当月最后一个工作的17:30之后自动生成打印报告。

③ 应该为每一个诊所生成一个报告，并列出各种药的药名、处方的总数、药物总量，以及所开药物的成本。

④ 如果药物可以分为不同剂量单位（如10mg，20mg），就要为每一剂量单位单独生成报告。

⑤ 对所有成本报告的访问权限应只限于管理访问控制单上的授权用户。

内容提要

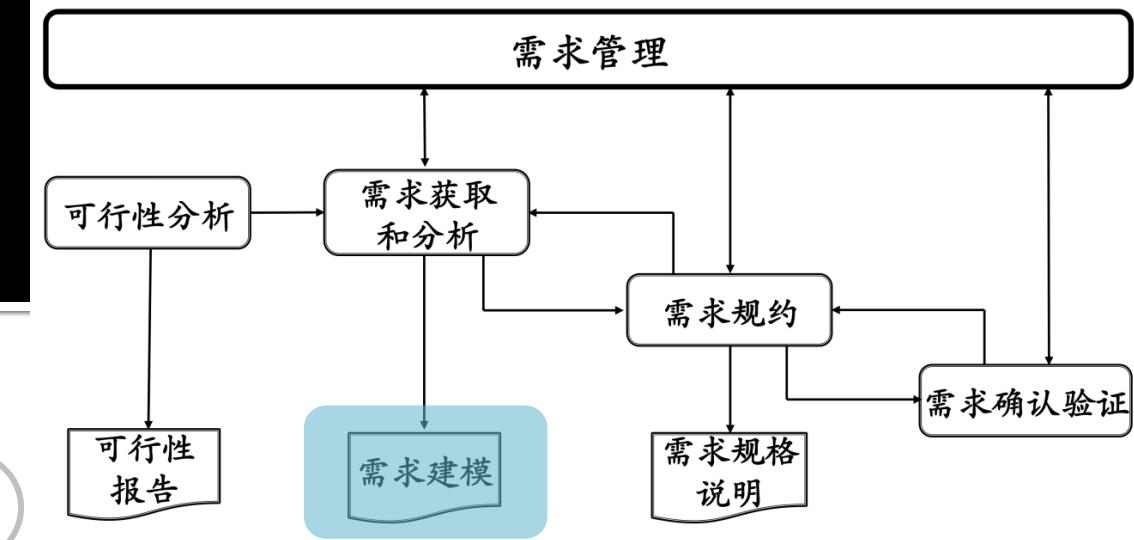
基本概念

需求获取

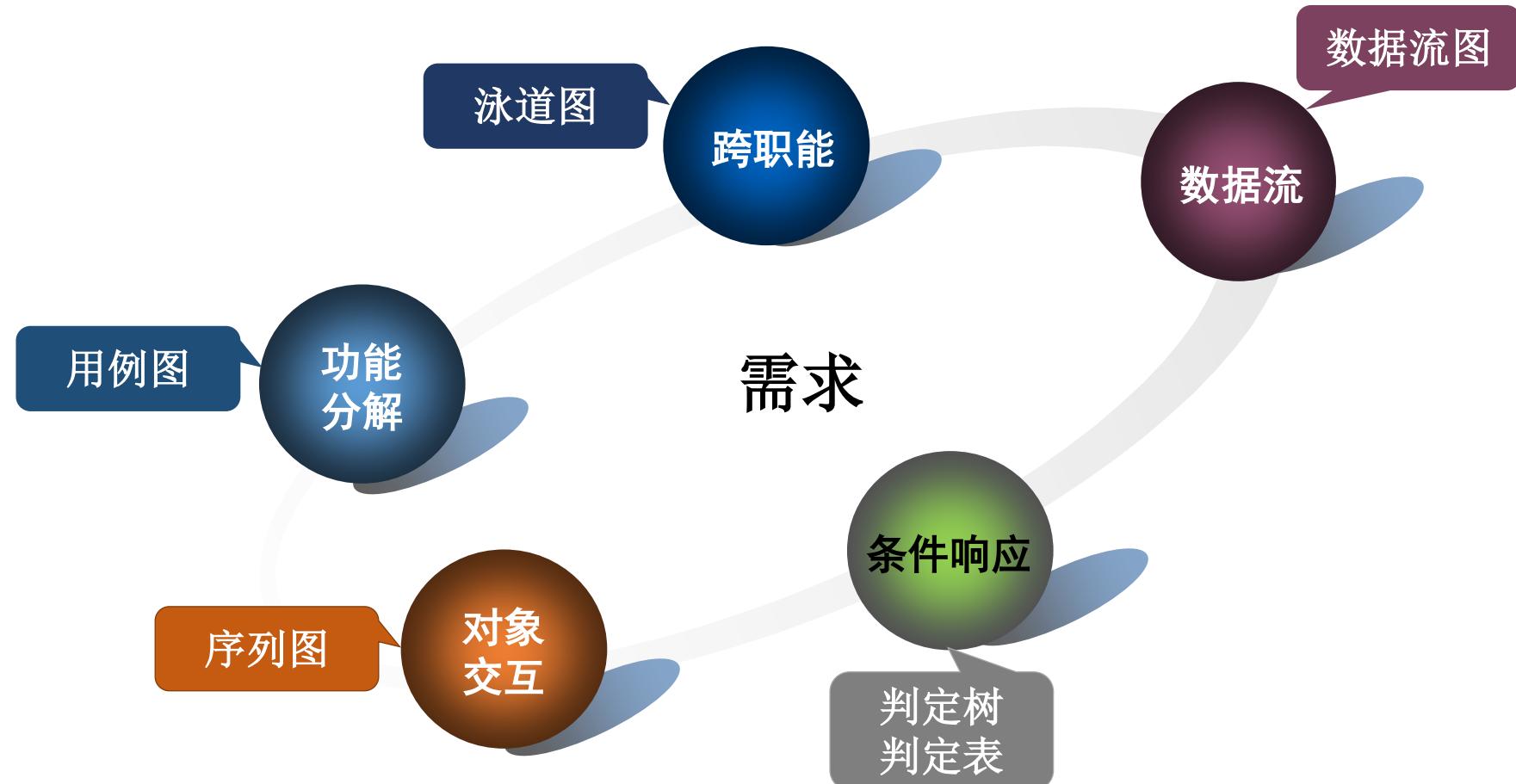
需求建模

需求规约

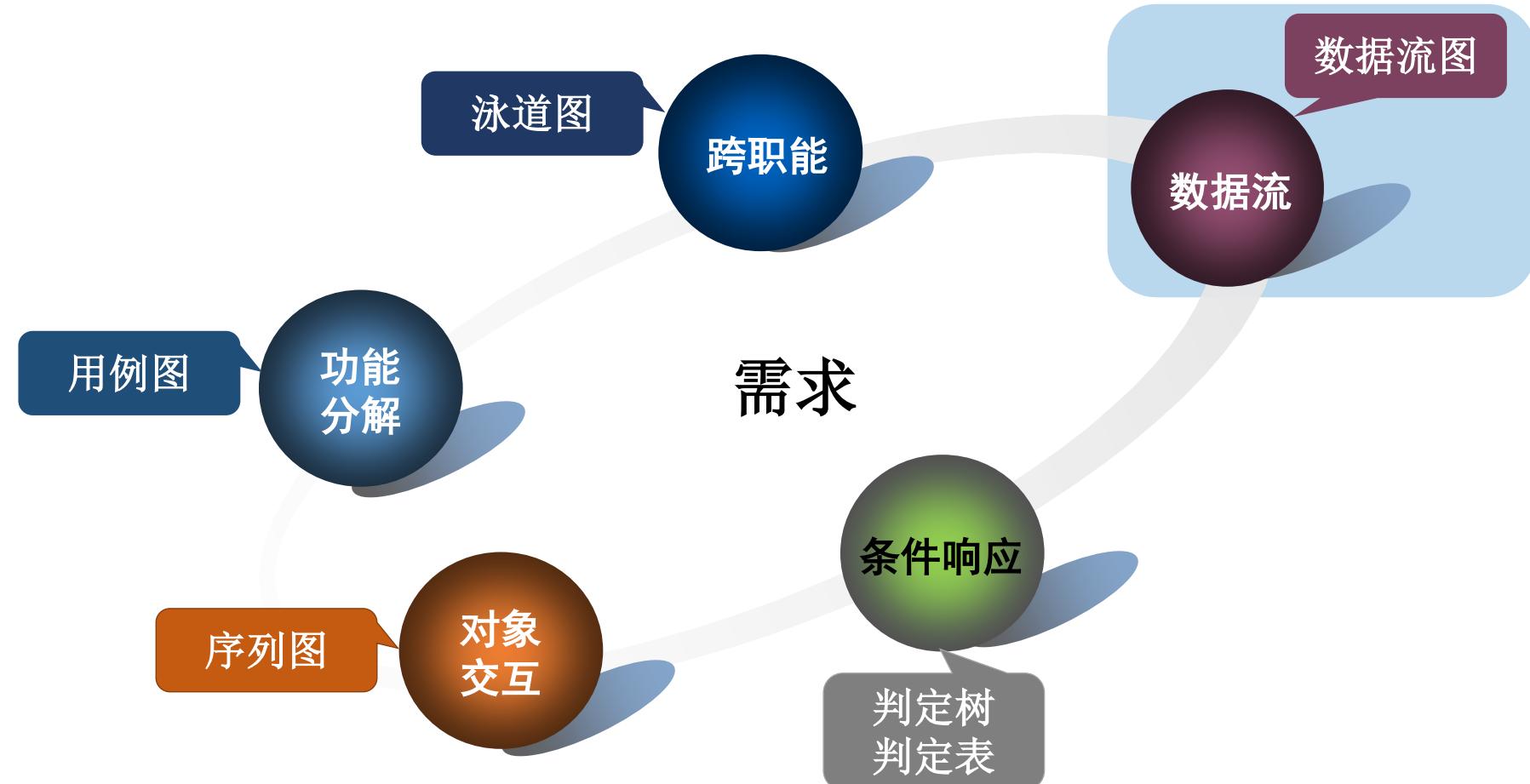
需求管理



需求建模



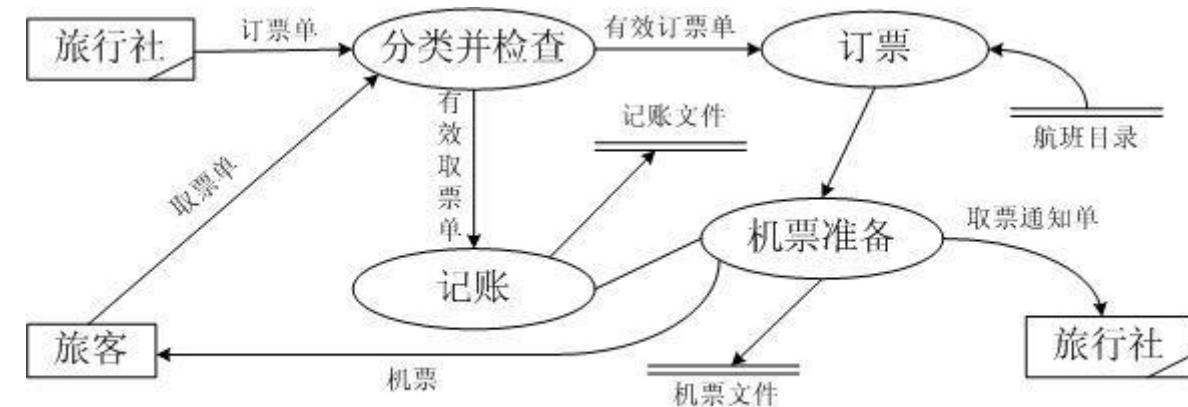
数据流图



数据流图

- 标识系统中的数据流动、处理和存储的过程
- 展示了数据流经系统的全貌，特别适合用于事务处理系统和其他偏重功能的系统，一般配合流程图一同使用

- 数据流：用箭头表示；
- 加工：用圆或椭圆表示；
 - 用简洁用语（动词加名词）命名
 - 单个数据流图中加工不要超过10个
- 数据存储：用双杠表示；
- 外部实体：用方框表示



泳道图



我们需要构建一个能够跟踪化学制剂的信息系统，这个系统应该能够跟踪我们仓库和实验室里所有的化学制剂。这样一来，药剂师就能够使用其他人剩余的化学制剂而不是总是去购买新的。这会为我们节省大量经费。与此同时，健康与安全部门希望这个系统能够帮助他们大大减少向政府提供化学品使用和处理报告的工作量。你们能在五个月内及时开发出符合这些要求的系统吗？

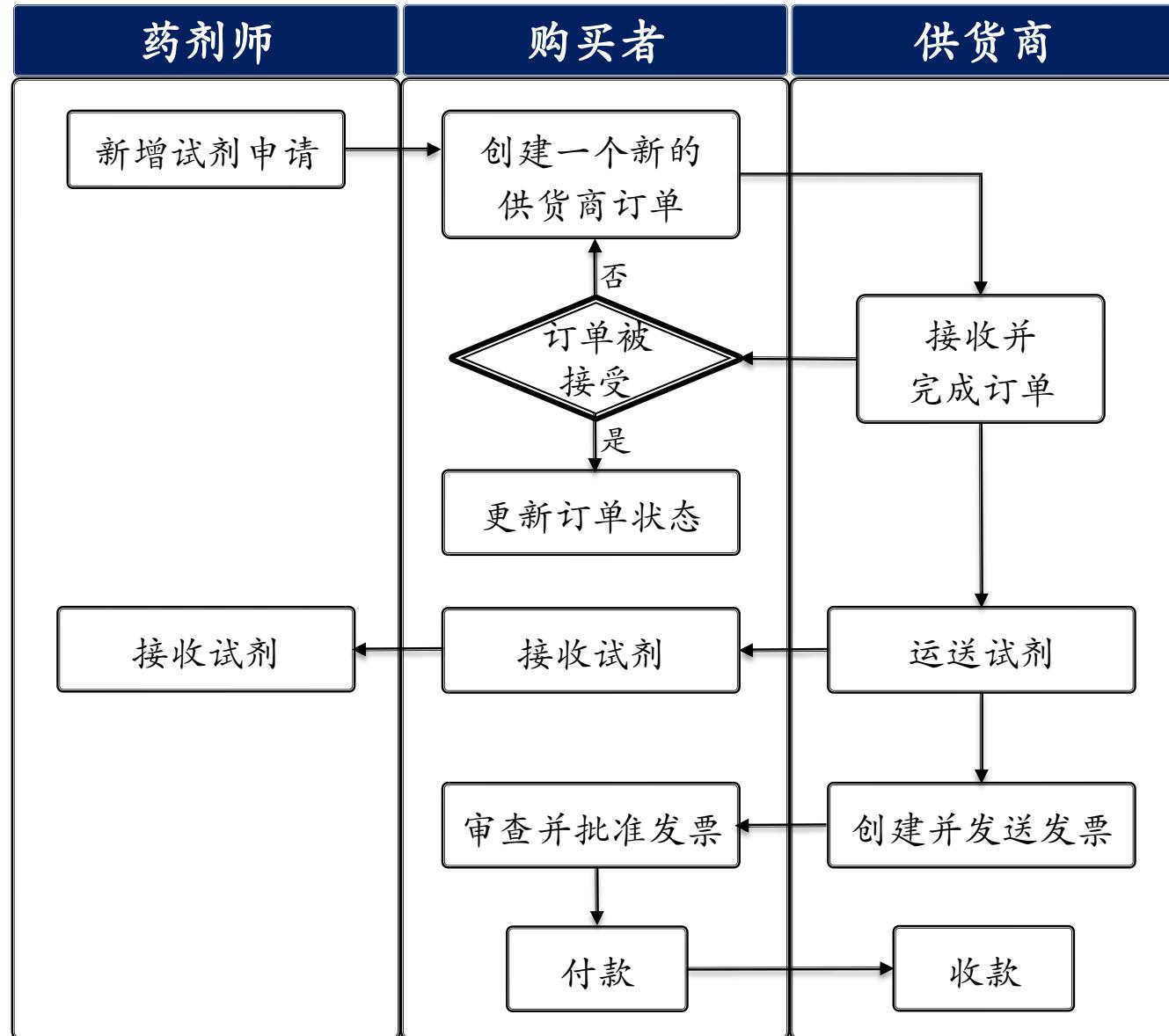


药剂师都很忙，他们没时间在你们开发之前确定所有的细节。你们的人难道就不能自己想想究竟该做什么吗？

我已经告诉了你需求了。现在就请开工吧。把你们的进度通报给我。

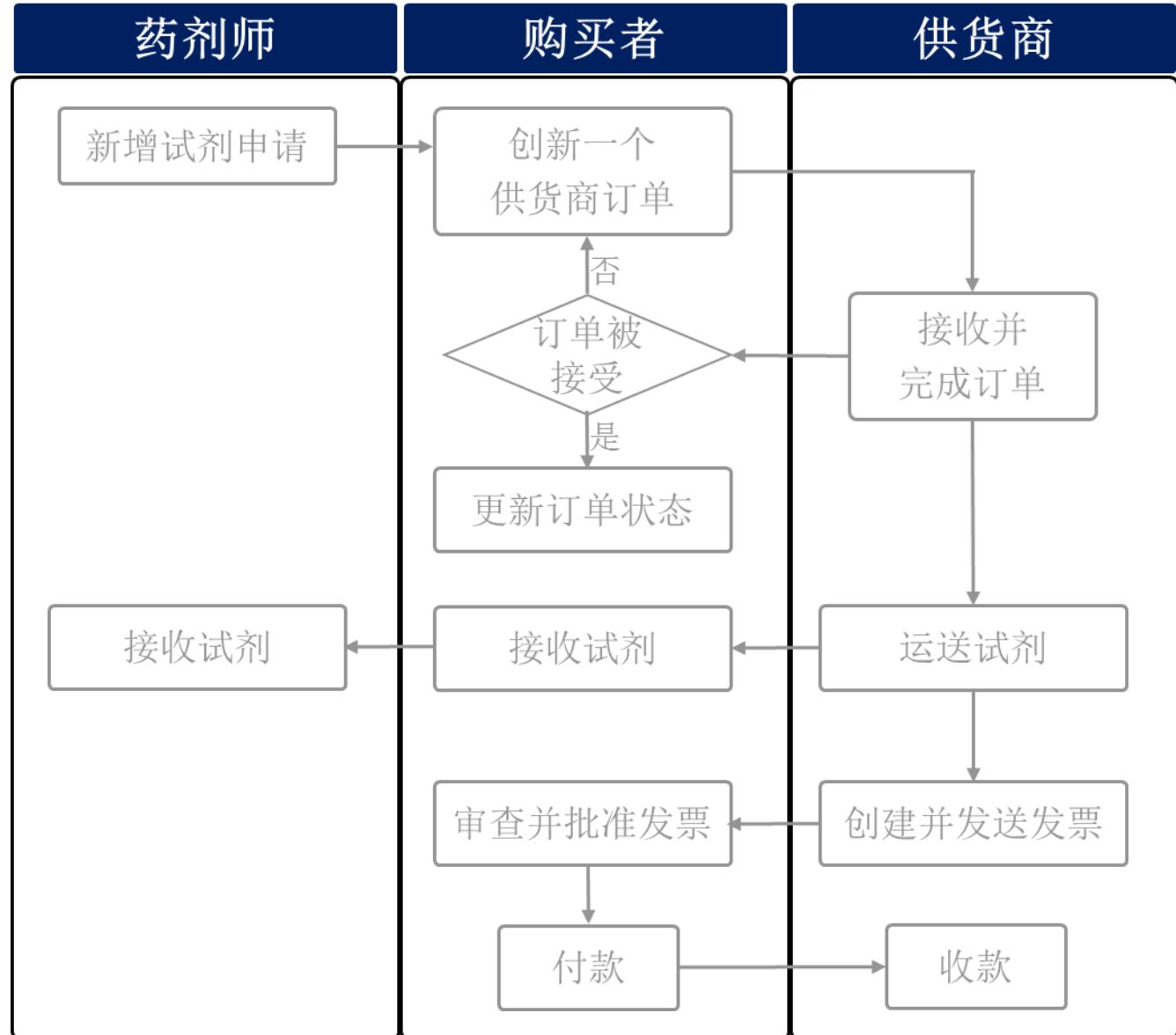
泳道图

- 描述业务过程中涉及的交互过程和操作步骤。



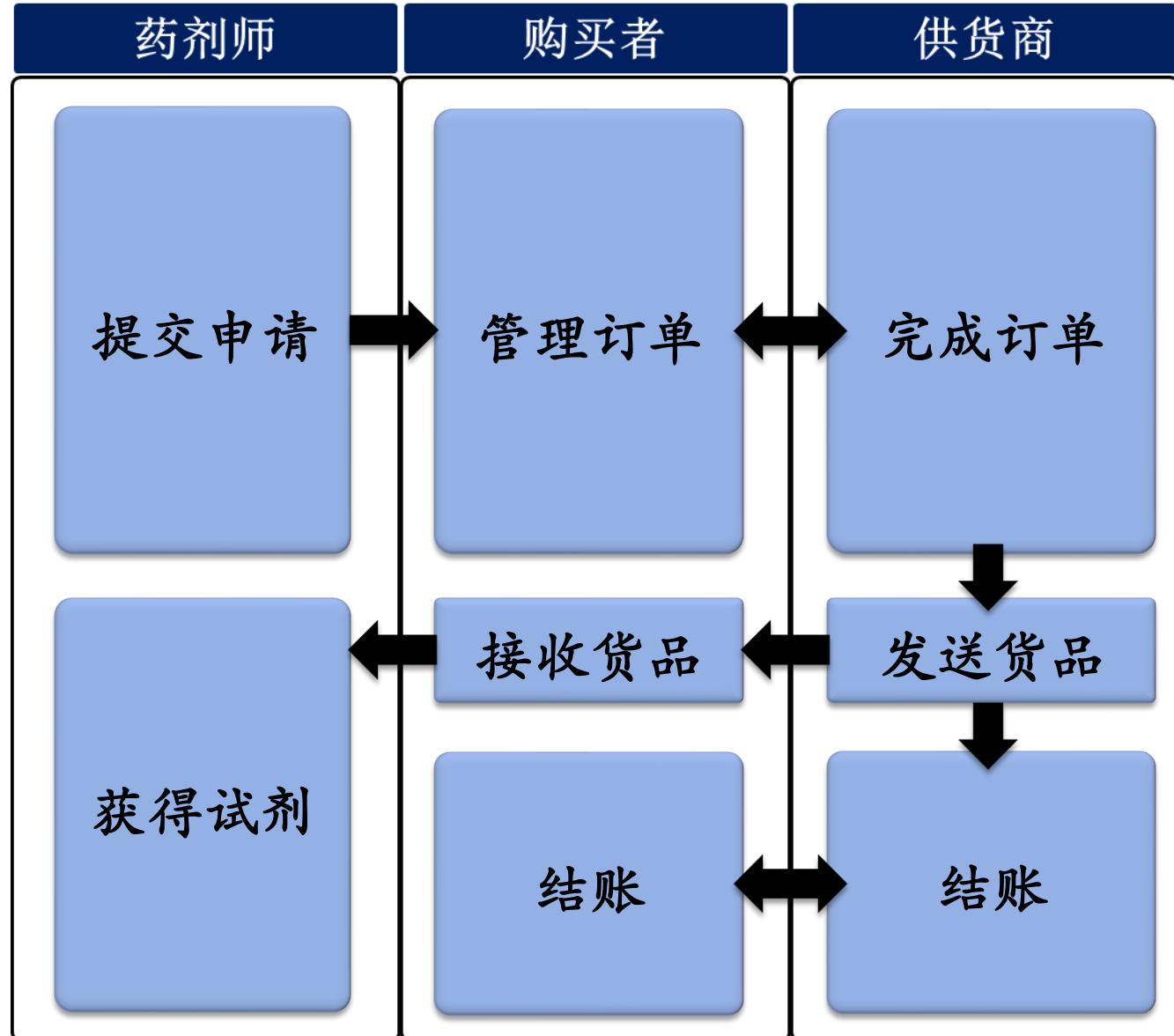
泳道图（跨职能流程图）

- 描述业务过程中涉及的步骤或软件系统的操作
- 整个系统分解为可视化的泳道，表示在流程中执行操作的不同子系统或执行者。
 - 分割流程的泳道
 - 流程步骤
 - 流程步骤之间的事务
 - 条件判定



1

分割流程的泳道

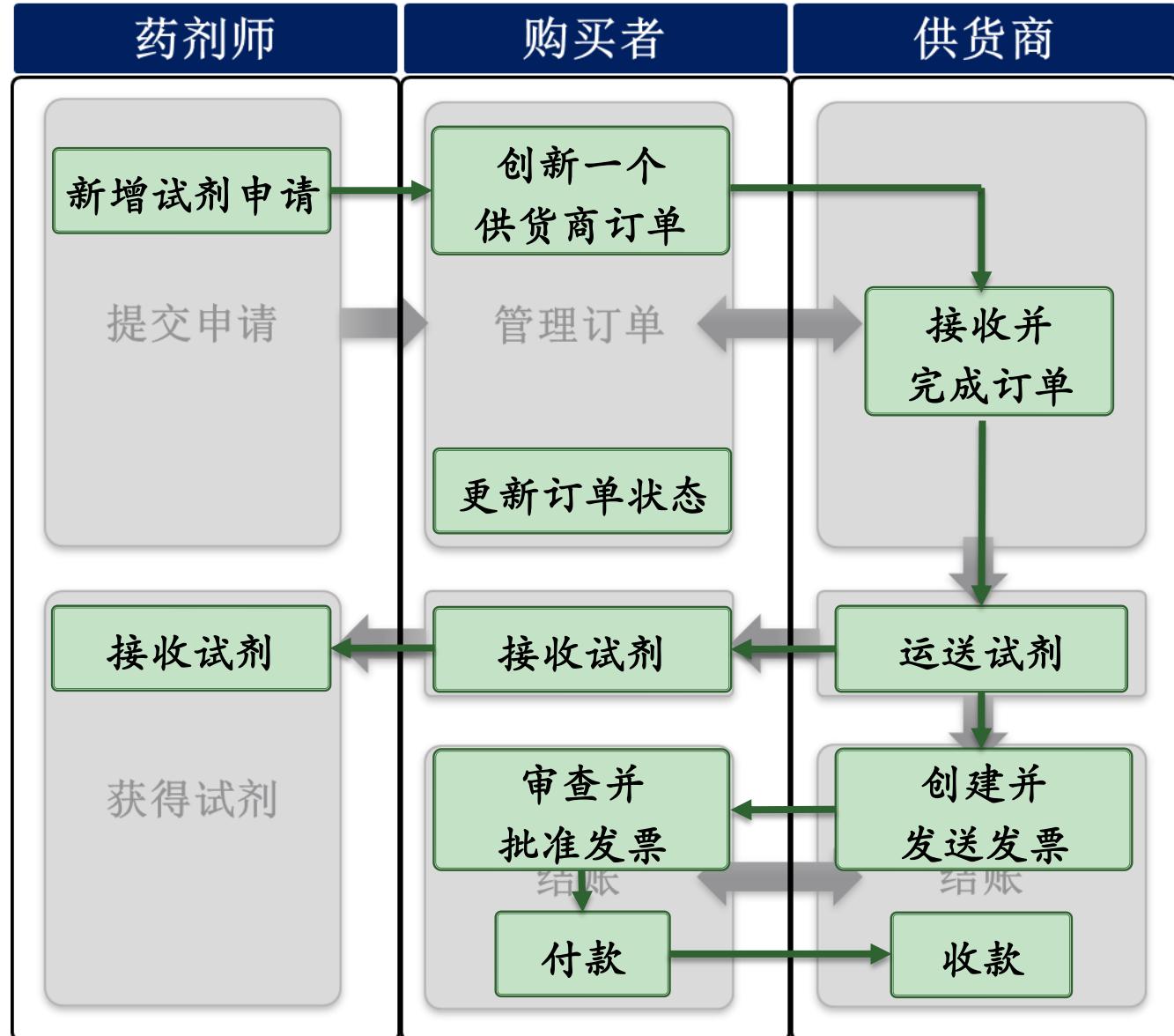


1

分割流程的泳道

2

流程



1

分割流程的泳道

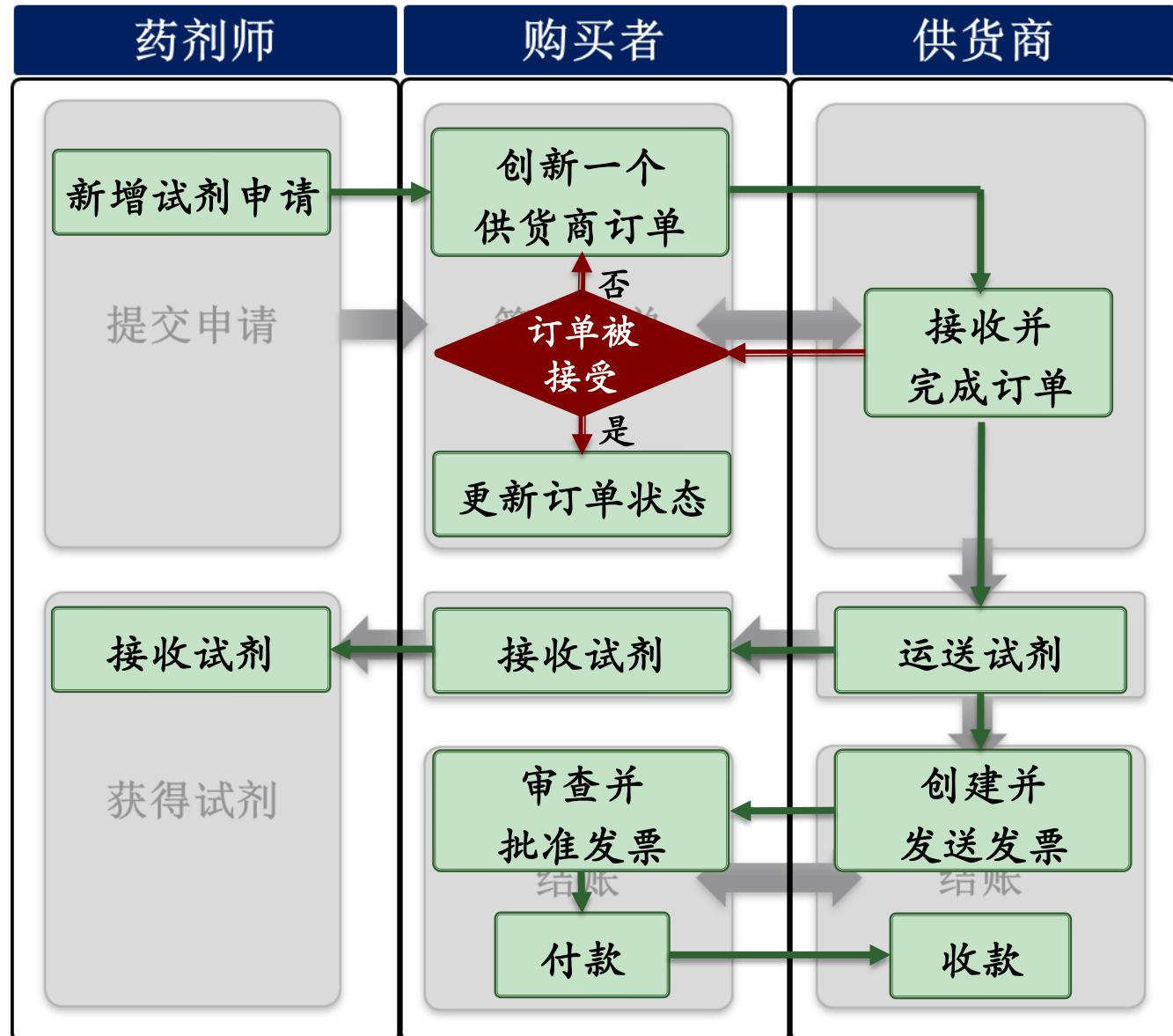
2

流程

操作步骤

角色之间的协同

事务



1

分割流程的泳道

2

流程

操作步骤

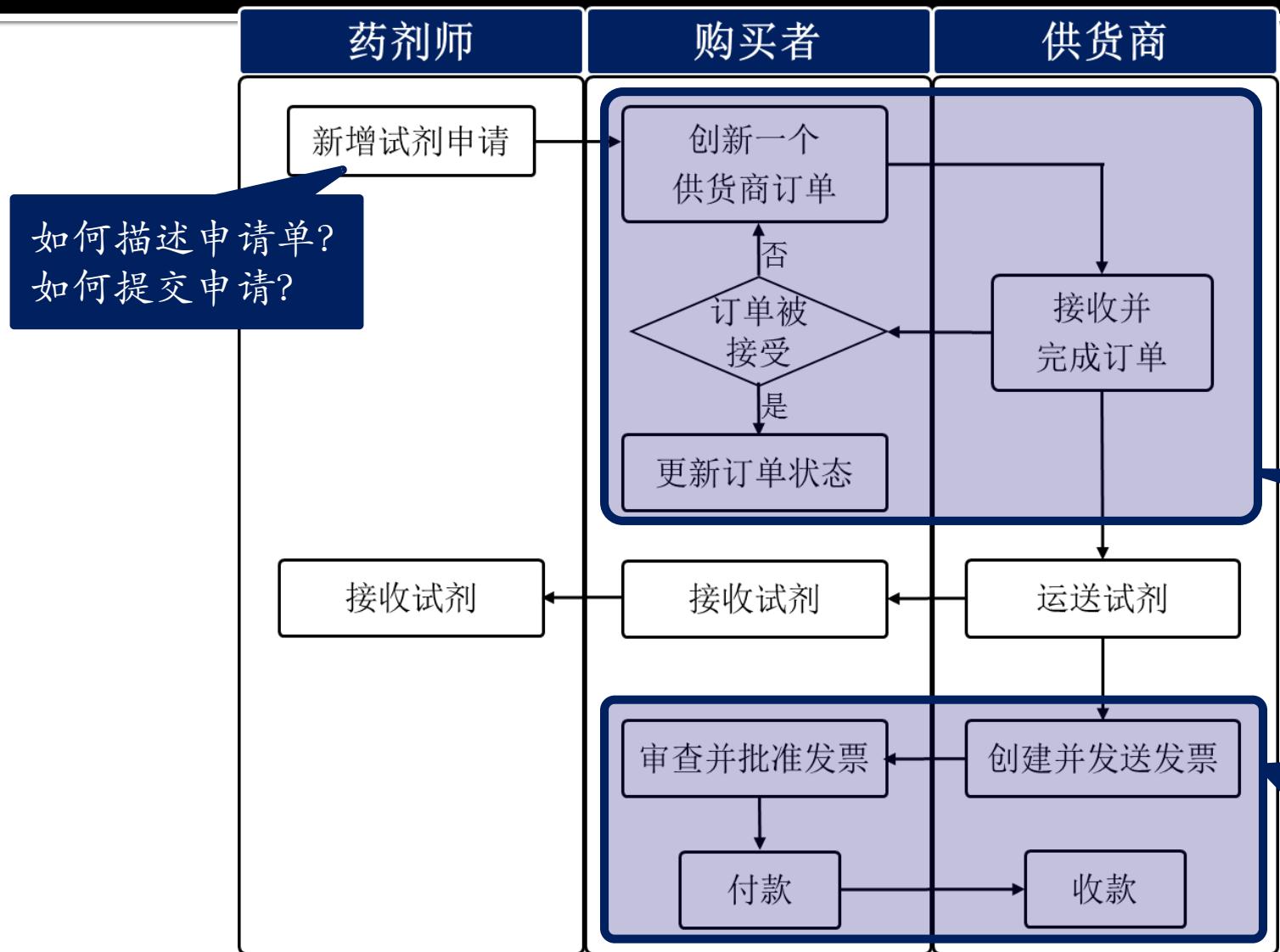
角色之间的协同

事务

3

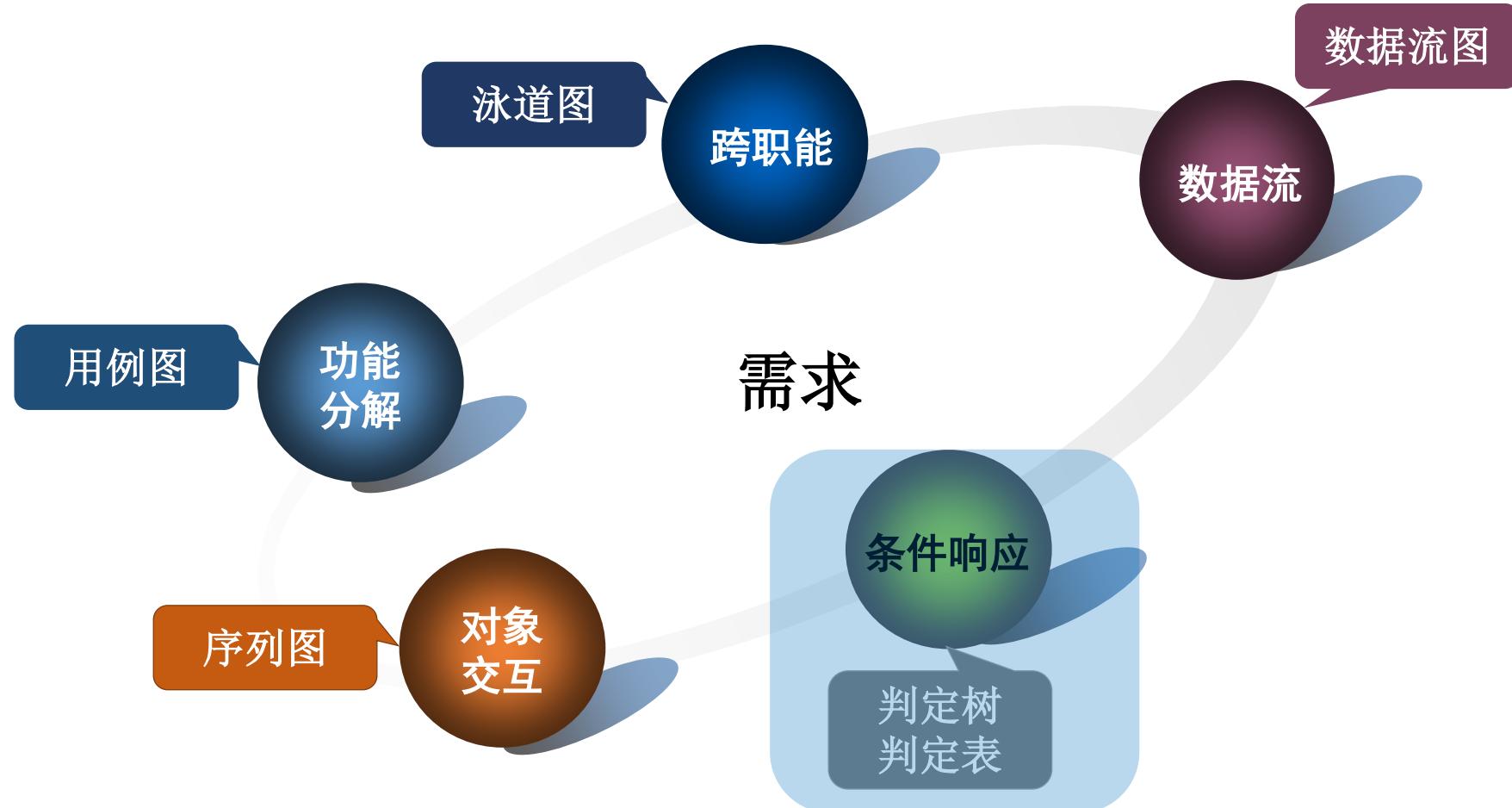
判定条件

业务过程的“草图”



2013-08-05 14:16:49	已收件
2013-08-05 15:31:13	快件在 上海 ,准备送往下一站 上海集散中心
2013-08-05 17:40:19	快件在 上海集散中心 ,准备送往下一站 上海集散中心
2013-08-05 22:52:09	快件在 上海集散中心 ,准备送往下一站 北京集散中心
2013-08-06 23:06:48	快件在 北京集散中心 ,准备送往下一站 廊坊集散中心
2013-08-07 06:23:06	快件在 廊坊集散中心 ,准备送往下一站 廊坊
2013-08-07 08:40:22	正在派件..
2013-08-07 09:11:38	派件已签收
2013-08-07 09:11:59	签收人是: 王

判定树和判定表



判定表和判定树

- **条件组合**: 软件系统经常受制于复杂的逻辑，有很多不同的条件组合，导致系统产生不同的行为。功能需求是在所有可能条件组合下对系统响应动作的描述，忽略条件组合将导致需求遗漏。
- 系统的逻辑和判定条件变得复杂之后，判定表和判定树有助于辅助分析和描述“条件组合—系统响应行为”之间的对应关系。
 - 列出影响系统行为的所有因素的各个取值，并且表明在每一个因素组合条件下系统预期的响应动作。

案例分析

化学品跟踪系统中，针对每个新的申请，是批准申请还是驳回申请，取决于以下四个因素：

1. 提出申请的用户是否有申请的权限。
2. 化学品库房或者外部供应商是否有货。
3. 该化学品是否在有毒化学品名单里，该名单所列的化学品要求有专门的安全操作培训。
4. 提出申请的用户是否已经接受过该类有毒化学品的操作培训。

判定树

化学品跟踪系统中，针对每个新的申请，是否批准申请或是驳回申请，取决于以下四个因素：

1. 提出申请的用户是否有申请的权限。
2. 化学品库房或者外部供应商是否有货。
3. 该化学品是否在有毒化学品名单里，该名单所列的化学品要求有专门的安全操作培训。
4. 提出申请的用户是否已经接受过该类有毒化学品的操作培训。

提出申请

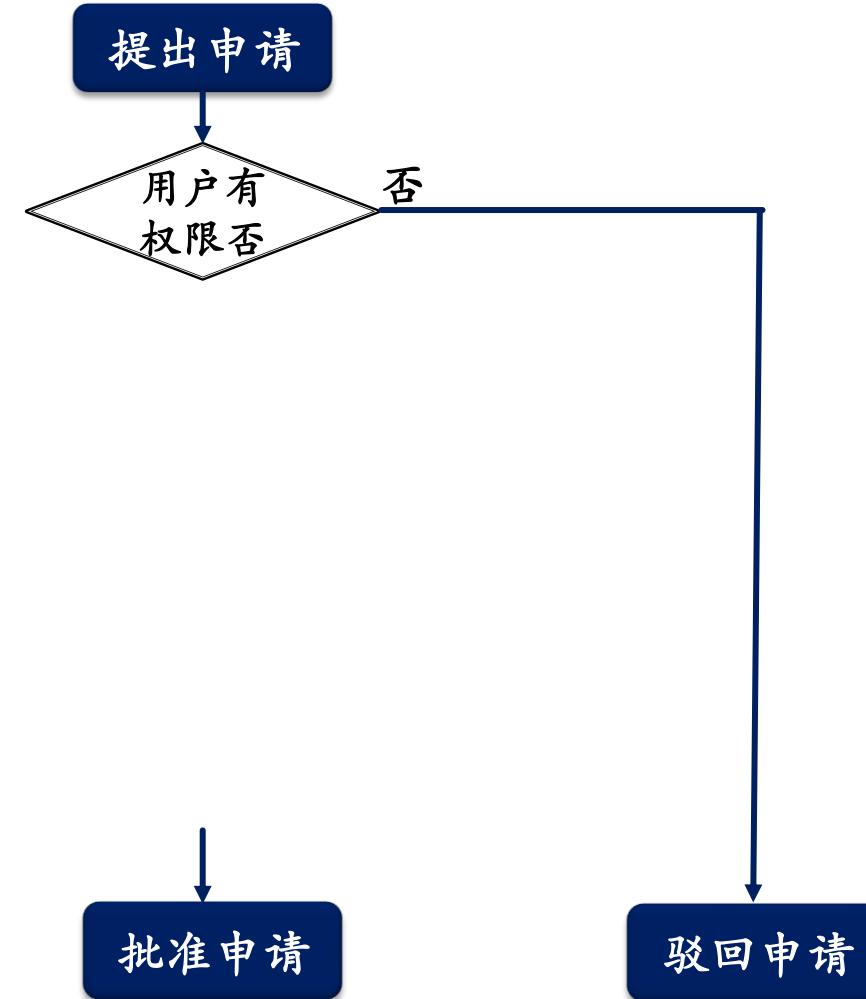
批准申请

驳回申请

判定树

化学品跟踪系统中，针对每个新的申请，是否批准申请或是驳回申请，取决于以下四个因素：

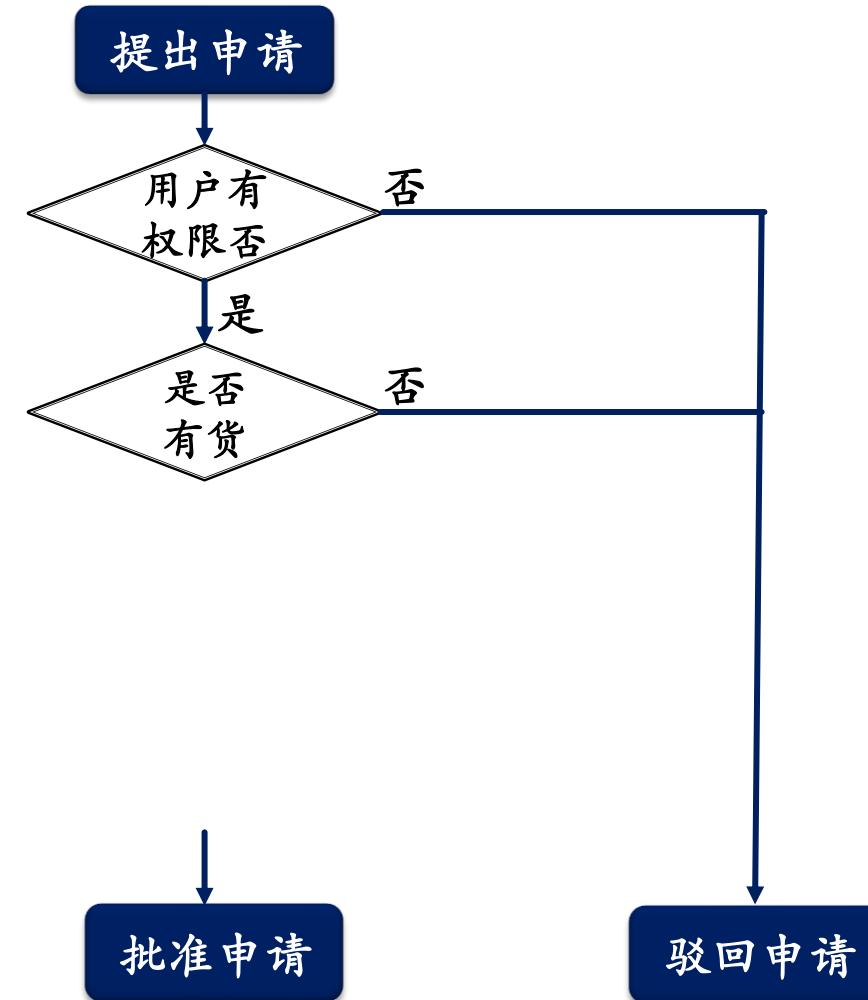
1. 提出申请的用户是否有申请的权限。
2. 化学品库房或者外部供应商是否有货。
3. 该化学品是否在有毒化学品名单里，该名单所列的化学品要求有专门的安全操作培训。
4. 提出申请的用户是否已经接受过该类有毒化学品的操作培训。



判定树

化学品跟踪系统中，针对每个新的申请，是否批准申请或是驳回申请，取决于以下四个因素：

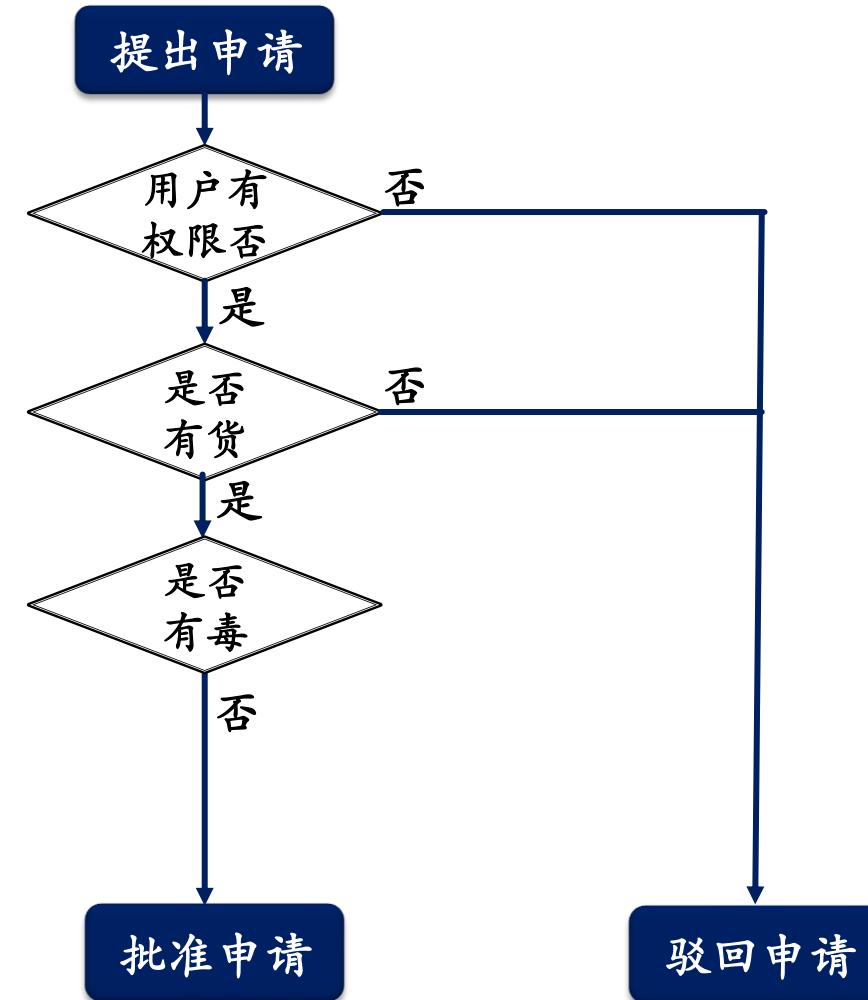
1. 提出申请的用户是否有申请的权限。
2. 化学品库房或者外部供应商是否有货。
3. 该化学品是否在有毒化学品名单里，该名单所列的化学品要求有专门的安全操作培训。
4. 提出申请的用户是否已经接受过该类有毒化学品的操作培训。



判定树

化学品跟踪系统中，针对每个新的申请，是否批准申请或是驳回申请，取决于以下四个因素：

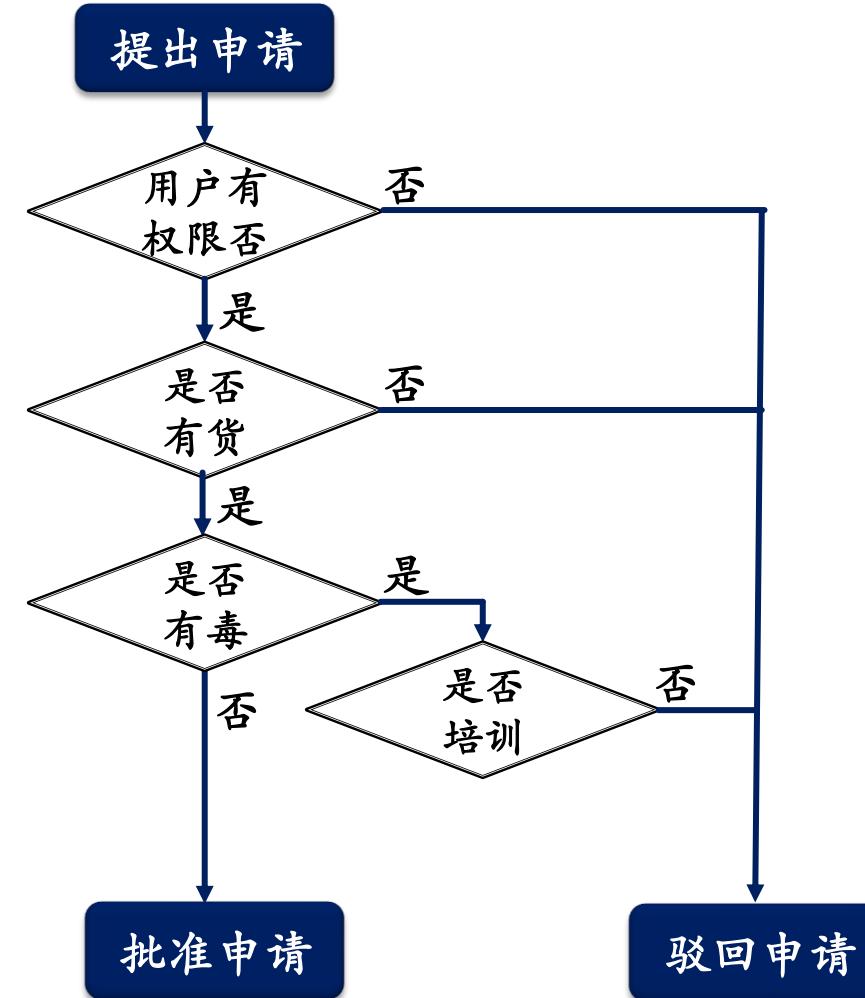
1. 提出申请的用户是否有申请的权限。
2. 化学品库房或者外部供应商是否有货。
3. 该化学品是否在有毒化学品名单里，该名单所列的化学品要求有专门的安全操作培训。
4. 提出申请的用户是否已经接受过该类有毒化学品的操作培训。



判定树

化学品跟踪系统中，针对每个新的申请，是否批准申请或是驳回申请，取决于以下四个因素：

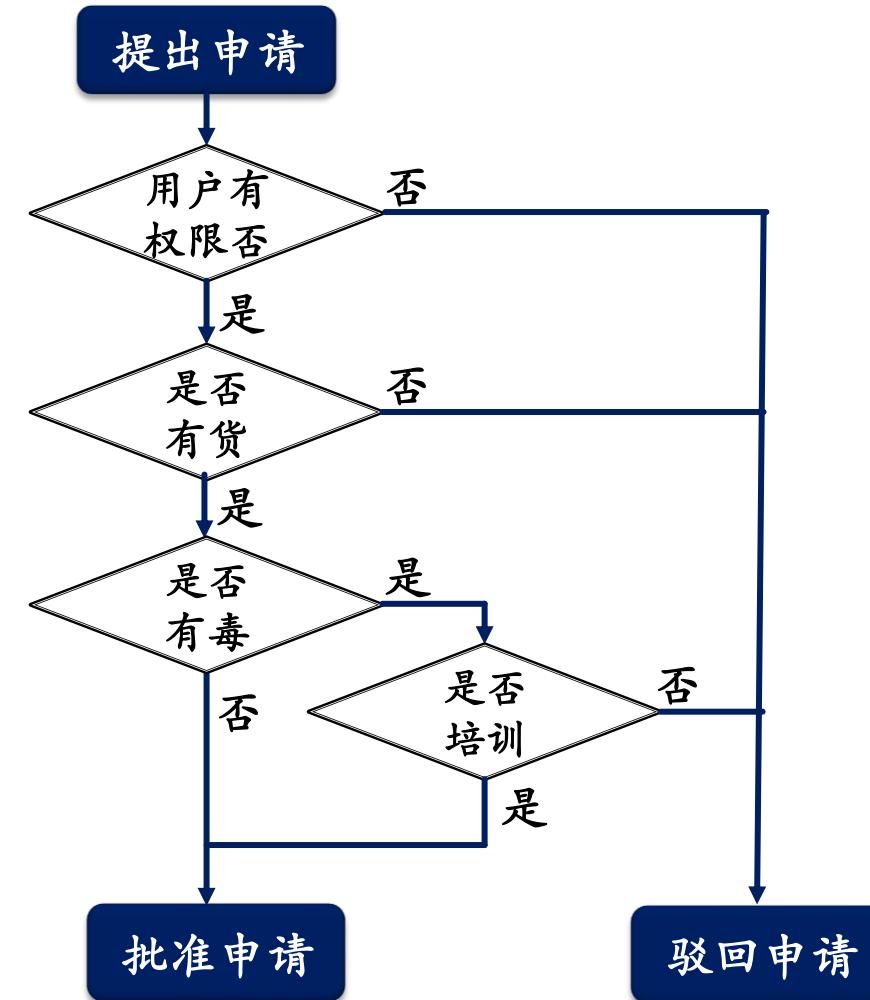
1. 提出申请的用户是否有申请的权限。
2. 化学品库房或者外部供应商是否有货。
3. 该化学品是否在有毒化学品名单里，该名单所列的化学品要求有专门的安全操作培训。
4. 提出申请的用户是否已经接受过该类有毒化学品的操作培训。



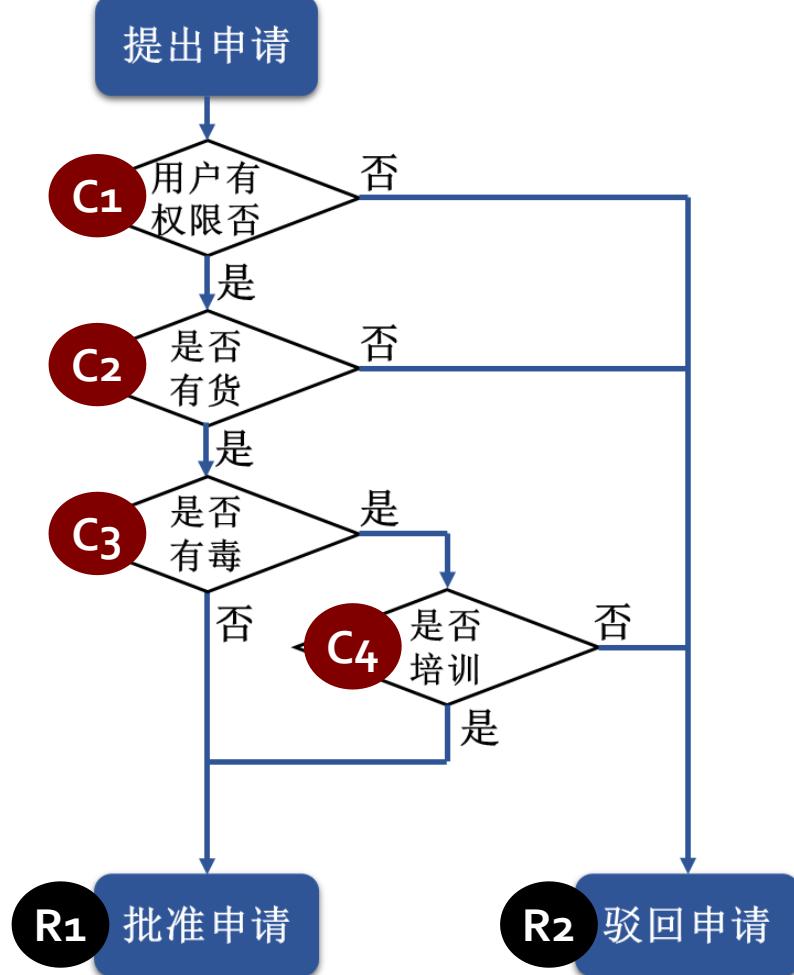
判定树

化学品跟踪系统中，针对每个新的申请，是否批准申请或是驳回申请，取决于以下四个因素：

1. 提出申请的用户是否有申请的权限。
2. 化学品库房或者外部供应商是否有货。
3. 该化学品是否在有毒化学品名单里，该名单所列的化学品要求有专门的安全操作培训。
4. 提出申请的用户是否已经接受过该类有毒化学品的操作培训。

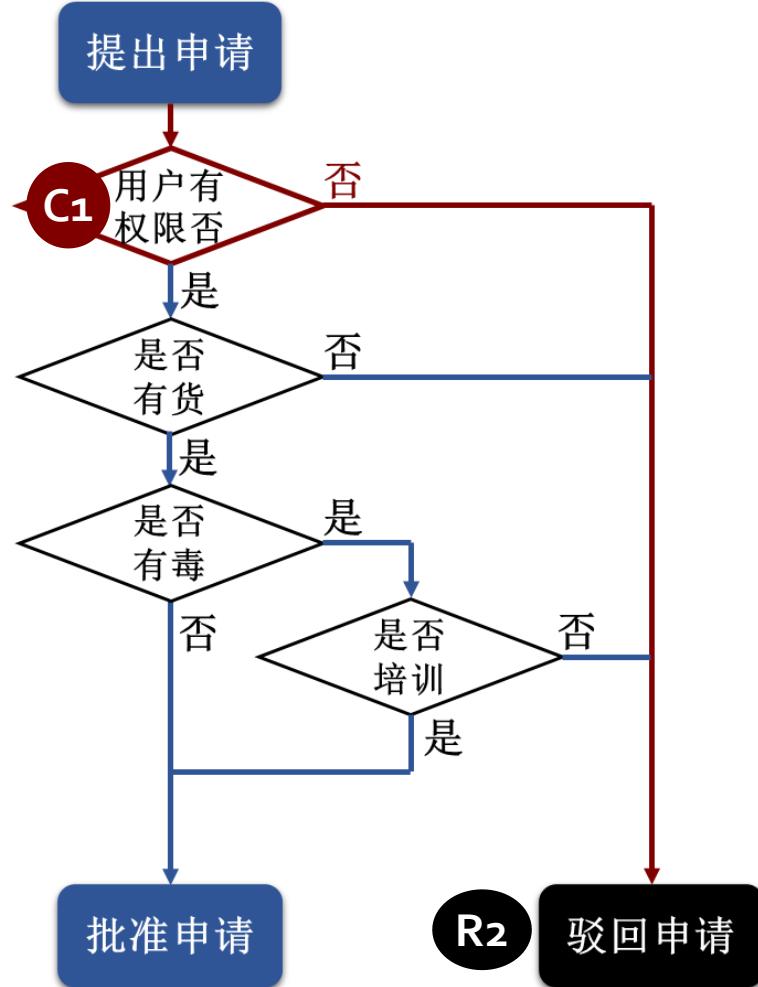


判定表



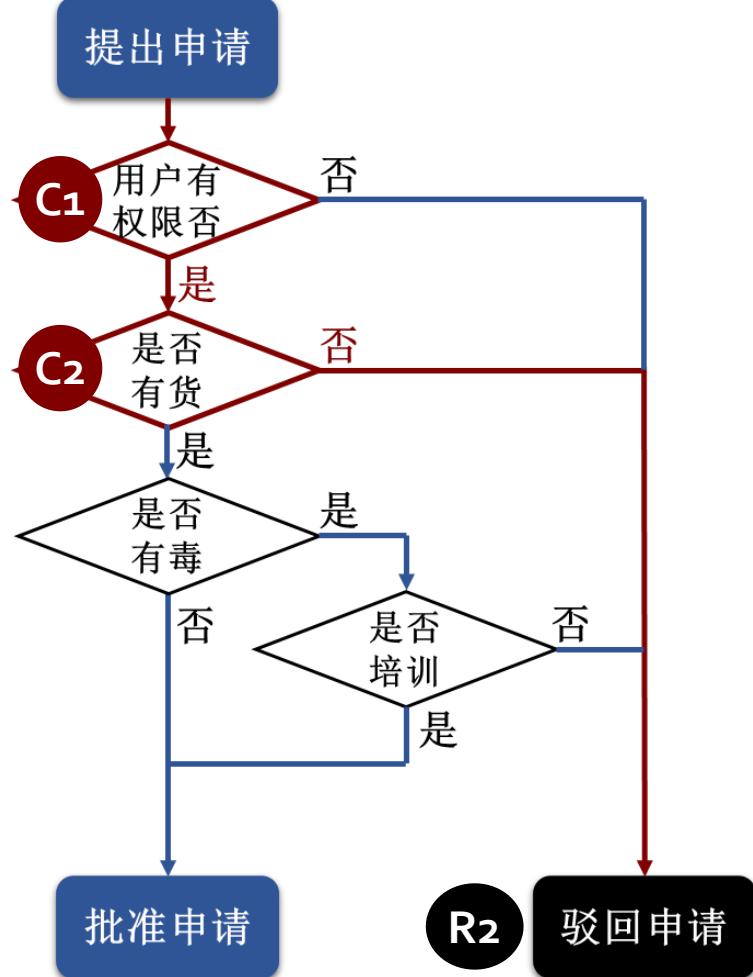
条件					
C1 用户有权限					
C2 有货					
C3 化学品有毒					
C4 申请人受过培训					
动作					
R1 批准申请					
R2 驳回申请					

判定表



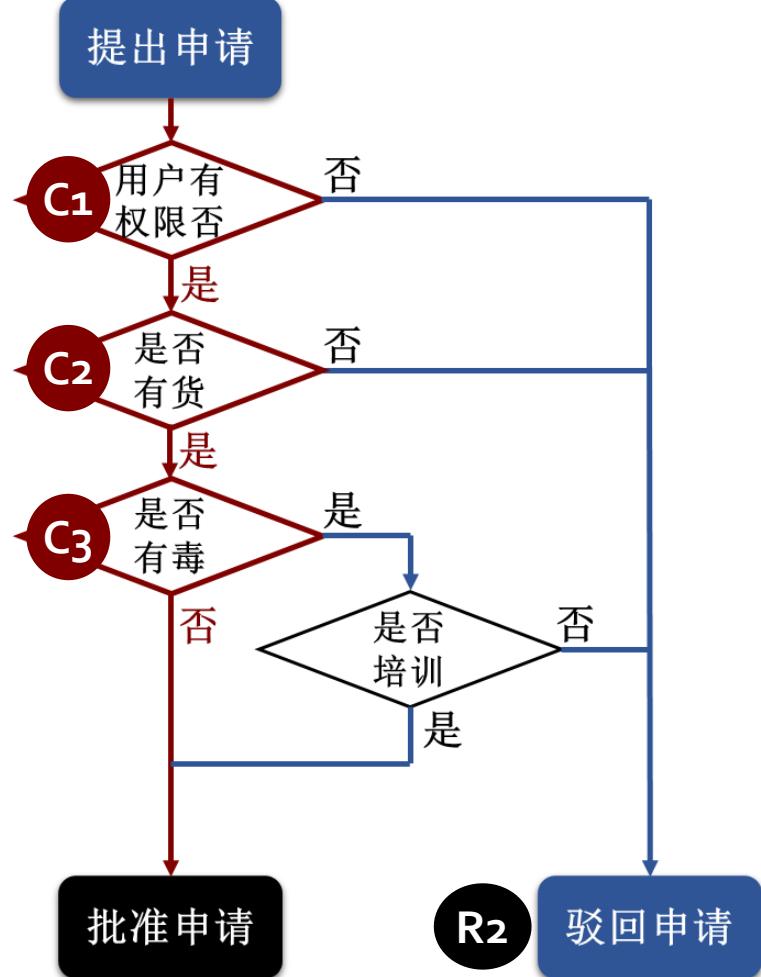
条件	1				
C1 用户有权限	F				
C2 有货	--				
C3 化学品有毒	--				
C4 申请人受过培训	--				
动作					
R1 批准申请					
R2 驳回申请	X				

判定表



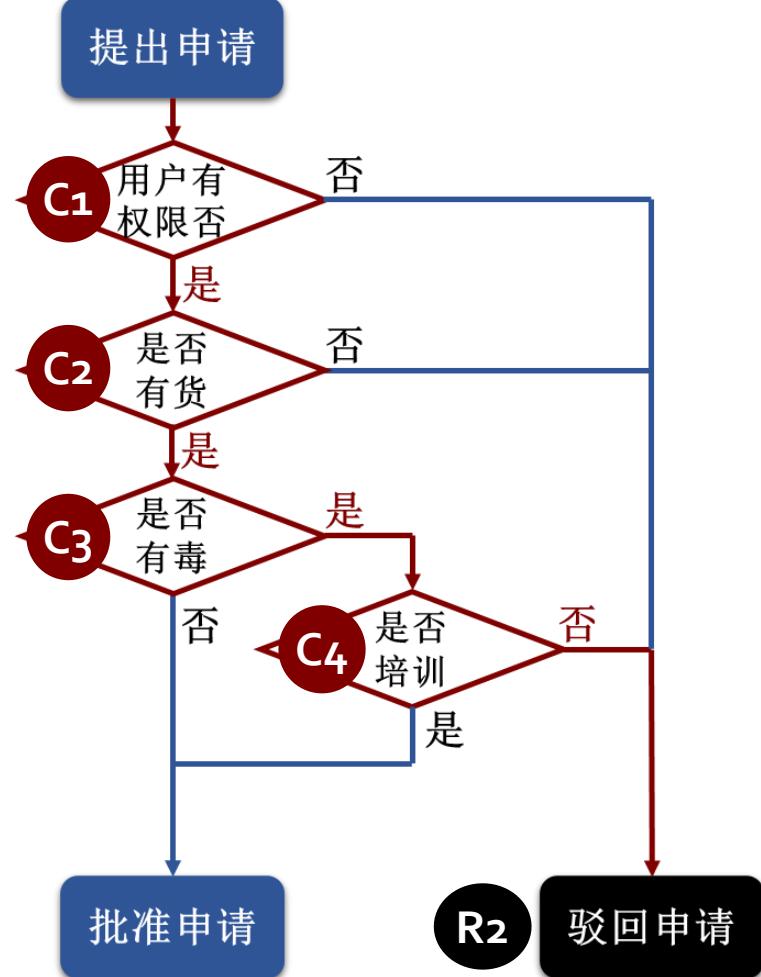
条件	1	2			
C1 用户有权限	F	T			
C2 有货	--	F			
C3 化学品有毒	--	--			
C4 申请人受过培训	--	--			
动作					
R1 批准申请					
R2 驳回申请	X	X			

判定表



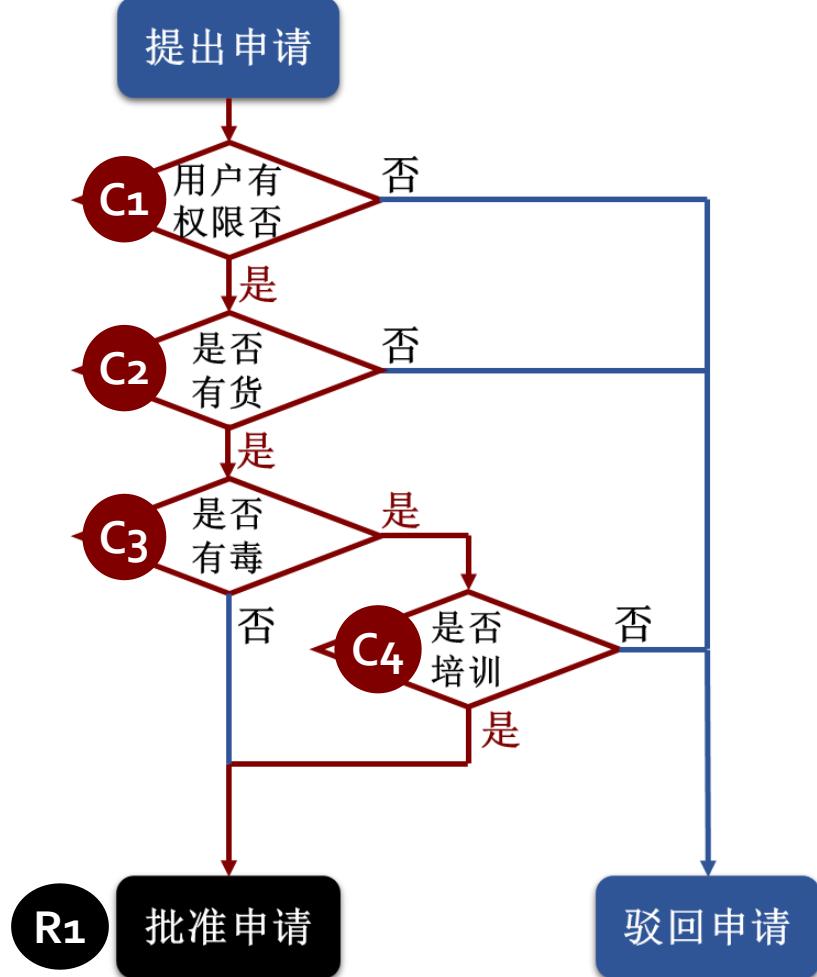
条件	1	2	3		
C1 用户有权限	F	T	T		
C2 有货	--	F	T		
C3 化学品有毒	--	--	F		
C4 申请人受过培训	--	--	--		
动作					
R1 批准申请			X		
R2 驳回申请	X	X			

判定表



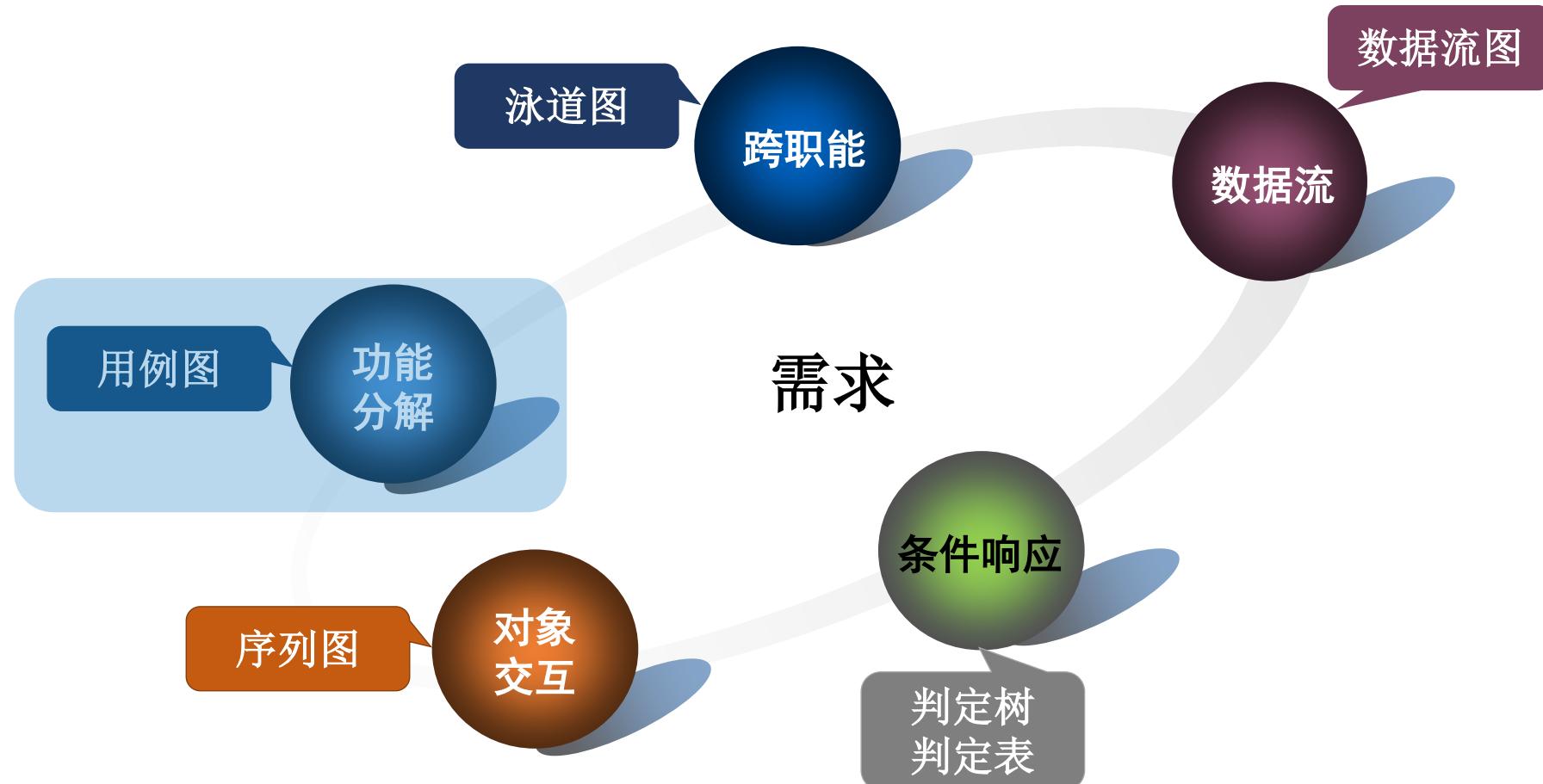
条件	1	2	3	4	
C1 用户有权限	F	T	T	T	
C2 有货	--	F	T	T	
C3 化学品有毒	--	--	F	T	
C4 申请人受过培训	--	--	--	F	
动作					
R1 批准申请			X		
R2 驳回申请	X	X		X	

判定表



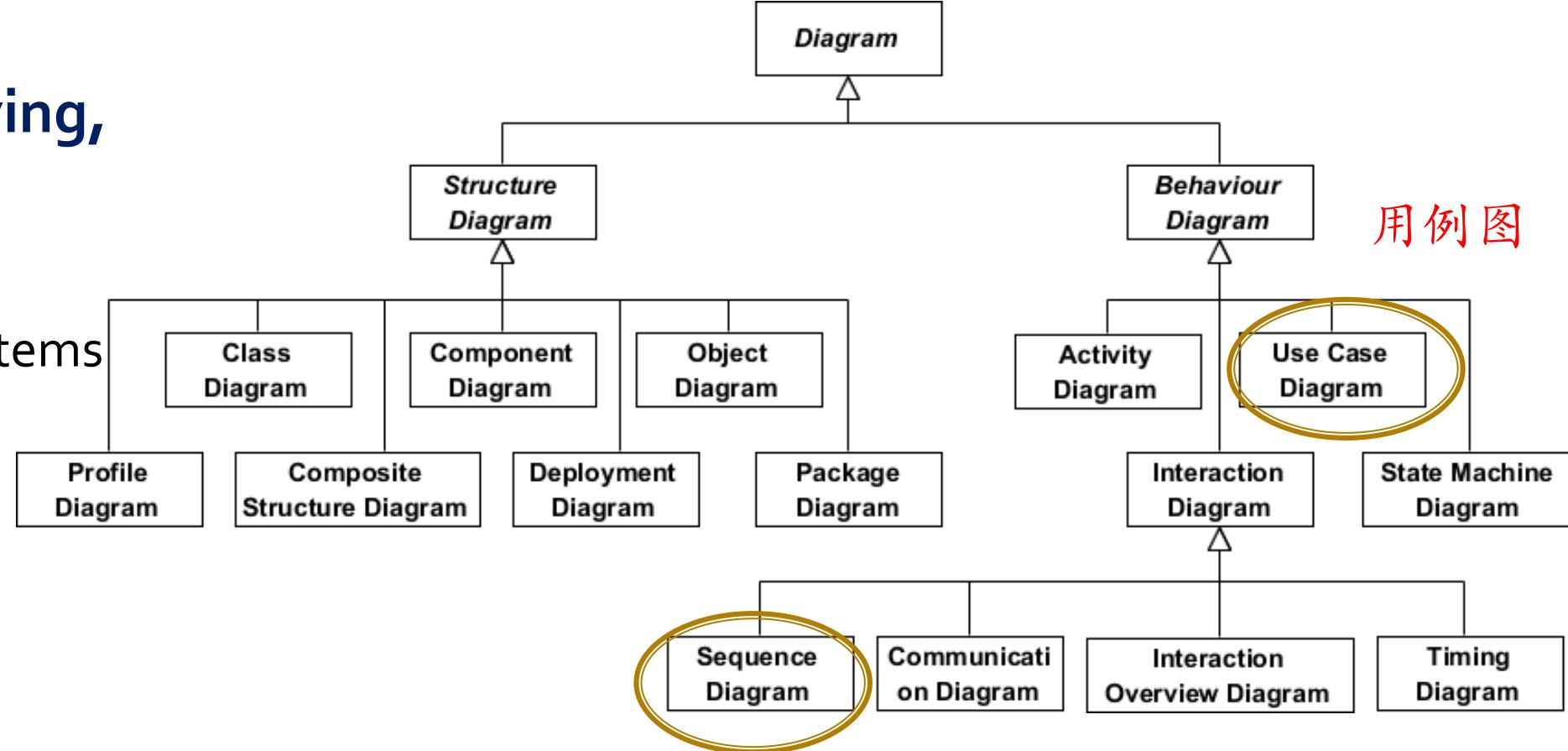
条件	1	2	3	4	5
C1 用户有权限	F	T	T	T	T
C2 有货	--	F	T	T	T
C3 化学品有毒	--	--	F	T	T
C4 申请人受过培训	--	--	--	F	T
动作					
R1 批准申请			X		X
R2 驳回申请	X	X		X	

用例图 Use Case Diagram in UML



Unified Modeling Language

UML is a graphical language for **Specifying, Visualizing, Constructing, Documenting** the artifacts of software systems



序列图

用例图

Grady Booch, Ivar Jacobson, James Rumbaugh, known as **the three amigos**

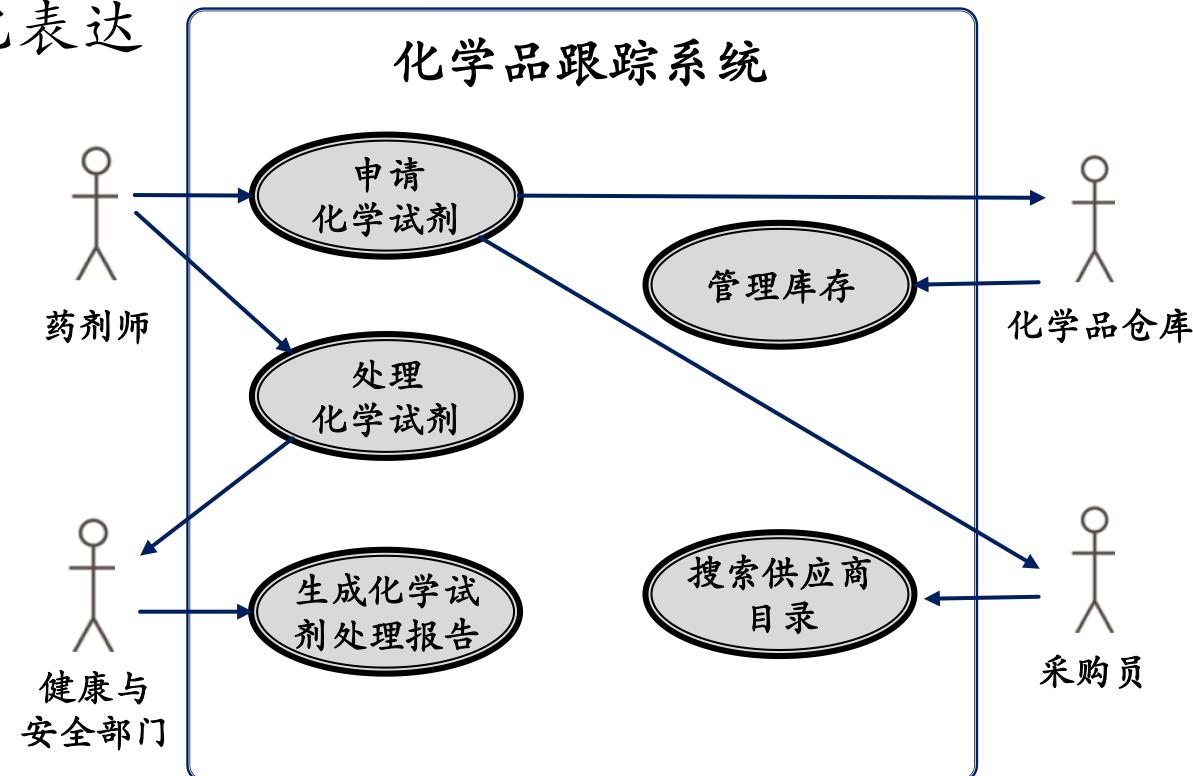
用例图 Use Case Diagram

- 用例图是对用户需求的一种概要性可视化表达



角色
与系统交互执行某个用例的人（或其他软件系统、硬件设备）。

用例
系统与某个外部角色之间的一系列交互。



用例不应该深入设计细节，
只要描述用户设想的交互活动就够了。

识别用例

我们需要构建一个能够跟踪化学制剂的信息系统，这个系统应该能够跟踪我们仓库和实验室里所有的化学制剂。这样一来，药剂师就能够使用其他人剩余的化学制剂而不是总去购买新的。这会为我们节省大量经费。与此同时，健康与安全部门希望这个系统能够帮助他们大大减少向政府提供化学品使用和处理报告的工作量。你们能在五个月内及时开发出符合这些要求的系统吗？

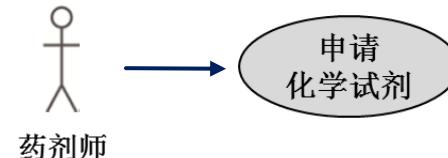


药剂师都很忙，他们没时间在你们开发之前确定所有的细节。你们的人难道就不能自己想想究竟该做什么吗？

我已经告诉了你需求了。现在就请开工吧。把你们的进度通报给我。

确定角色，分析角色和系统交互活动

药剂师 申请使用化学制剂



健康与安全部门 提交化学试剂使用报告



识别用例

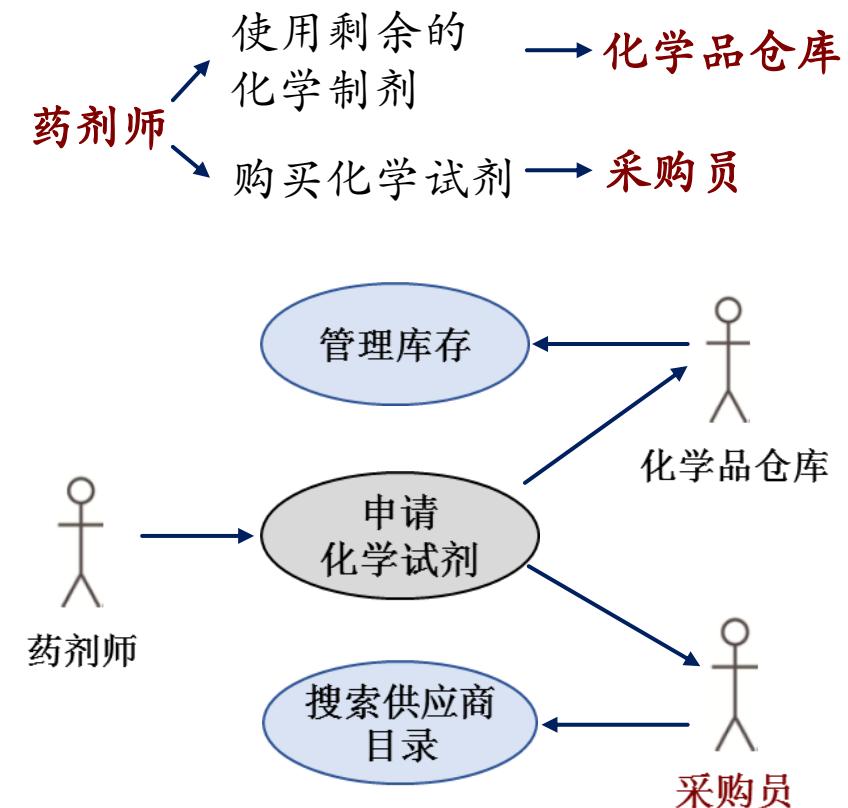
我们需要构建一个能够跟踪化学制剂的信息系统，这个系统应该能够跟踪我们仓库和实验室里所有的化学制剂。这样一来，药剂师就能够使用其他人剩余的化学制剂而不是总去购买新的。这会为我们节省大量经费。与此同时，健康与安全部门希望这个系统能够帮助他们大大减少向政府提供化学品使用和处理报告的工作量。你们能在五个月内及时开发出符合这些要求的系统吗？



药剂师都很忙，他们没时间在你们开发之前确定所有的细节。你们的人难道就不能自己想想究竟该做什么吗？

我已经告诉了你需求了。现在就请开工吧。把你们的进度通报给我。

分析业务过程，
将场景概括为
用例



识别用例

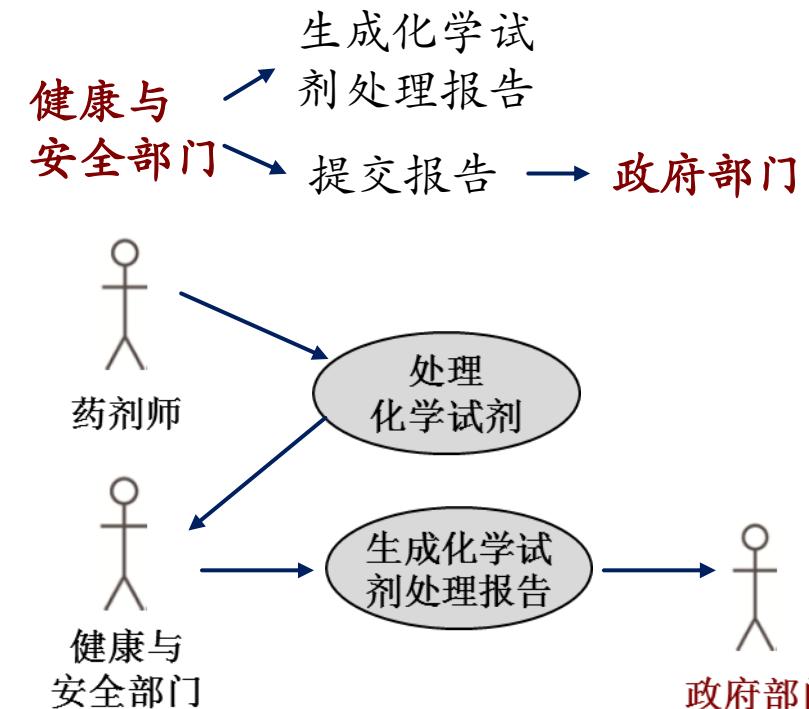
我们需要构建一个能够跟踪化学制剂的信息系统，这个系统应该能够跟踪我们仓库和实验室里所有的化学制剂。这样一来，药剂师就能够使用其他人剩余的化学制剂而不是总去购买新的。这会为我们节省大量经费。与此同时，健康与安全部门希望这个系统能够帮助他们大大减少向政府提供化学品使用和处理报告的工作量。你们能在五个月内及时开发出符合这些要求的系统吗？



药剂师都很忙，他们没时间在你们开发之前确定所有的细节。你们的人难道就不能自己想想究竟该做什么吗？

我已经告诉了你需求了。现在就请开工吧。把你们的进度通报给我。

识别系统外部事件，并关联到角色和用例



Generic Rules

- Actor
 - All the roles interacted with the system have been represented by Actors
 - Each actor is associated with at least one use case
 - Compose the actors if one actor includes the other
 - Decompose the actors if they are associated with different unrelated use cases
- Use Case
 - All the major functionalities of the system have been represented by use cases
 - Commonality analysis among use cases to identify new use cases and the extension, generalization and inclusion relationships among use cases
 - Decompose complex use cases

“误入歧途”

不要强行将每一个需求都“硬塞”到一个用例(相似的需求可以合并),也不要试图把所有的需求都以用例的方式表示出来(例如非功能需求就不用表示)。

“用例的合并”

用例描述

颗粒度

抽象层次

根据输入/输出关系区分功能,但用例说明中不应含有数据定义。

“用户的视角”

将用例关联到业务过程或目标。

用例
用户不
明白

用例
过多

用例
过于
复杂

用例
内含数
据定义

用例
内含
设计

如果一个流程超过10到15步,就要考虑分解为多个场景。

“用例的分解”



用例<不等于>用户界面
用例<不等于>系统结构
聚焦用户与系统之间的交互

用例场景

- 记录用例的基本信息，是图的**必要的补充说明**。基本信息包括：
 - 一个唯一的ID和一个简洁的名称（指明用户目标）
 - 一个简短的文字说明，用来描述用例的意图
 - 开始执行用例的触发条件
 - 用例开始需要满足的零个或多个前提条件
 - 一个或多个后置条件，描述用例成功完成后的系统状态
 - 一个有编号的步骤序列，展示了角色与系统之间的交互顺序

化学品跟踪系统	
药剂师	申请化学试剂
化学品仓库	管理库存
ID和名称:	UC-4申请化学品
创建人:	Lorj
创建日期:	2013年8月22日
首要角色:	申请发起人
次要角色:	买家, 化学品仓库, 培训数据库
描述:	申请人输入名称或化学品ID也可以通过导入结构化化学绘图工具来指定自己需要的化学品。系统也给申请人提供了从化学品仓库获取或是由申请人从供应商那订货的选择
触发条件:	申请人表示他需要一种化学品
前置条件:	1. 用户的身份通过认证 2. 用户被授权申请化学品 3. 化学品库存数据库在线
后置条件:	1. 申请被存储到CTS中 2. 申请被发送给化学品仓库或是某个买家
正常流程:	4.0 从化学品仓库申请化学品 1. 申请人指定所需的化学品 2. 系统列出化学品仓库中剩余的被申请化学品 3. 系统给申请人提供查看所有化学品容器历史的功能 4. 申请人选择某个特定的容器或是要求供应商订货(参见4.1) 5. 申请人输入其他信息来完成申请 6. 系统存储申请并通知化学品仓库
可选流程:	4.1 从供应商申请化学品 1. 化学的申请人搜索供应商目录(参见4.1.E1) 2. 系统显示供应商列表提供可用的大小、等级和价格的化学品 3. 申请人选择一个供应商, 容器大小、等级和容器的数量 4. 申请人输入其他信息来完成申请 5. 系统存储申请并通知买方
异常:	4.1 E1化学品非商用 1. 系统显示信息: 没有化学品供应商 2. 系统询问申请人他是想申请另一件化学品(3a)还是要退出(4a) 3a. 申请人要求申请另一件化学品 3b. 系统启动正常流程 4a. 申请人要求退出 4b. 系统终止用例
优先级:	高
使用频率:	每位药剂师每周大约5次, 化学品仓库人员每周200次
商业规则:	BR-28, BR-31
其他信息:	从所有支持化学品绘制的软件包中, 系统都可以用标准编码的形式导入一个化学品结构
假设:	导入的化学品结构假定都有效

用例：条件

申请
化学品

■ 触发条件

- 事件，引发相关用例的执行

触发
条件

申请人需要一种化学药品

■ 前置条件 定义系统开始执行用例之前必须满足的先决条件

- 描述系统的状态，例如：ATM系统的用例“提取现金”，必需满足前置条件“ATM机中有现金”。
- 系统在用例执行前，需检测满足所有的前置条件。

前置
条件

1. 用户的身份通过认证
2. 用户被授权申请化学品
3. 化学品库存数据库在线

■ 后置条件 描述用例执行成功后系统的状态

- 用户可观察到的内容（例如：系统显示帐户余额）
- 物理产出（例如：自动取款机吐钱并打印收据）
- 内容系统状态变化（例如：账户被扣除提现的金额与交易费用）

后置
条件

1. 申请被存储到CTS中
2. 申请被发送给化学品仓库或是某个买家

用例：流程

申请
化学品

正常流程

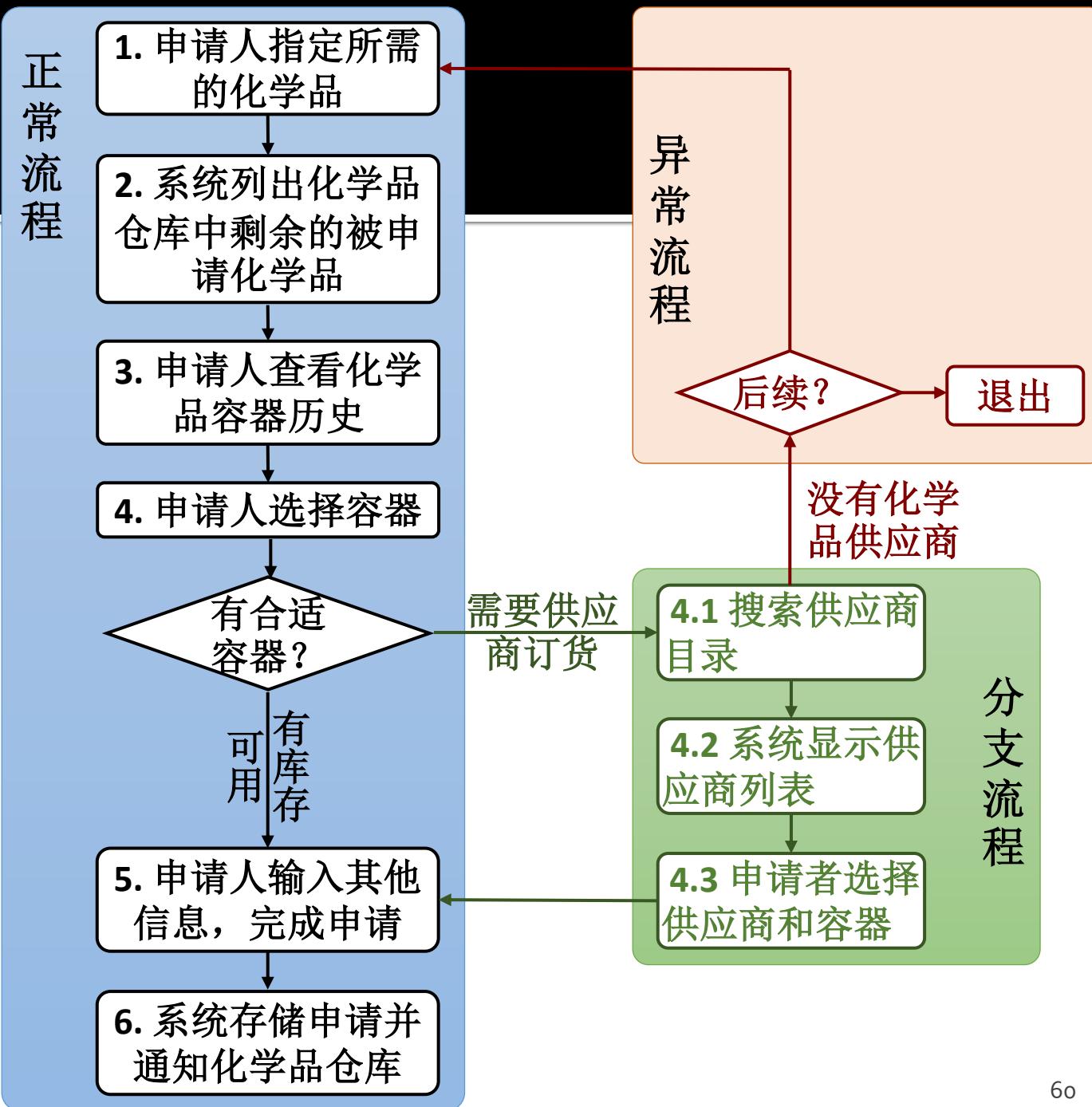
- 用例事件正常执行的步骤序列，缺省的行为
- 场景，又称主要流程、基本流程、快乐路径等

可选流程

- 用例内一些步骤可变，出现分支，但同样成功执行
- 二级场景

异常流程

- 不能成功执行的场景，描述错误条件及其处理方法



用例 vs 用户故事

■ 用户故事表达：卡片



验收标准

- 每10秒钟读取DecaWatt计量表的数据，在门户上显示15分钟的增量，.....



■ 良好的用户故事表达：INVEST



Independent 独立性



Negotiable 可协商



Valuable 有价值



Estimable 可估算



Small 小型



Testable 可测试

用例 vs 用户故事

User Story

Defines the who, what and why of a product feature

Use Case

Focuses on the functionalities of a feature or a process

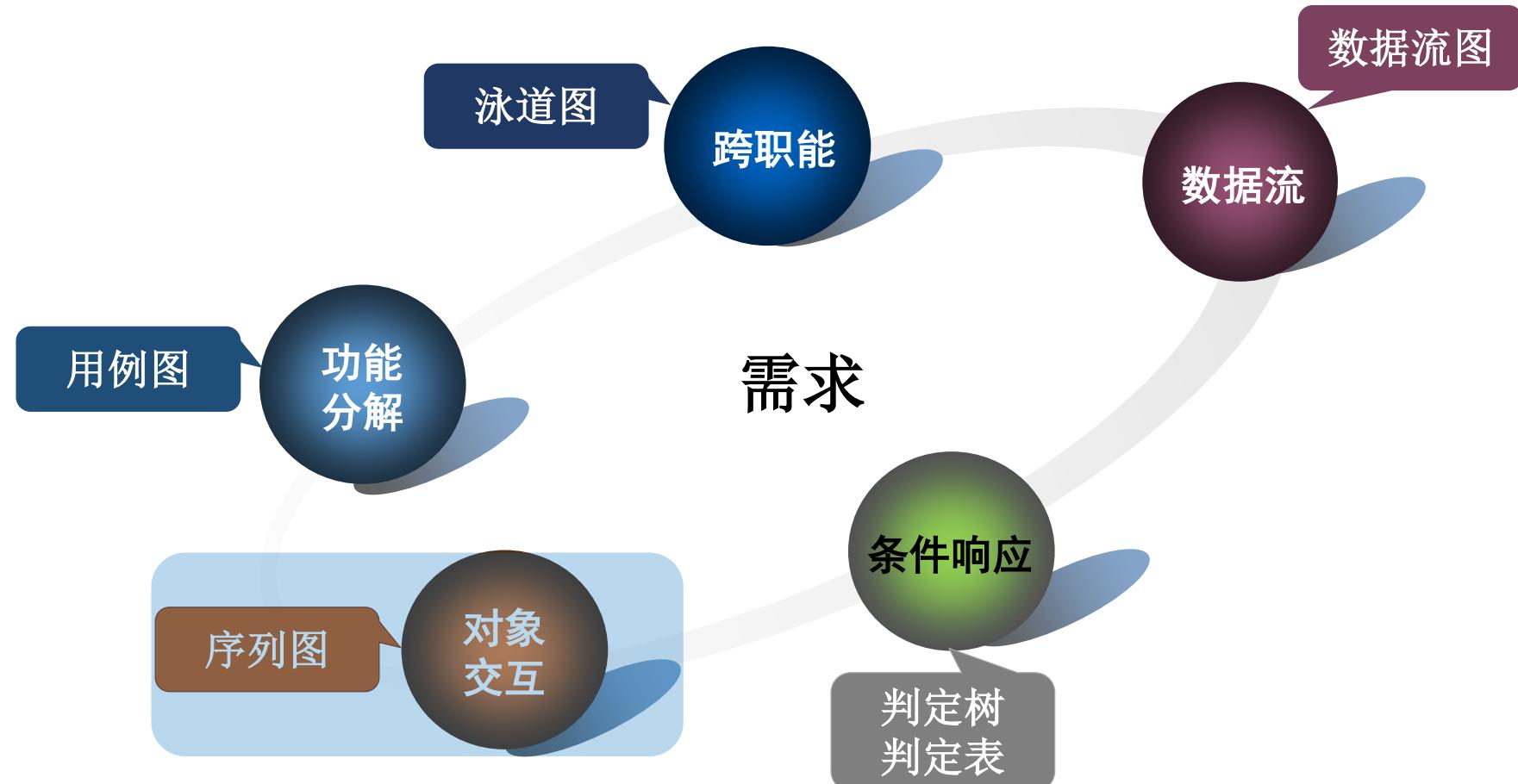
Use Story(用户故事)：描述系统对用户或客户有价值的功能，只是需求描述，而不是详细的需求规范

Use Case(用例)：在不展现一个系统或子系统内部结构的情况下，对系统或子系统的某个连贯的功能单元的定义和描述

用例 vs 用戶故事

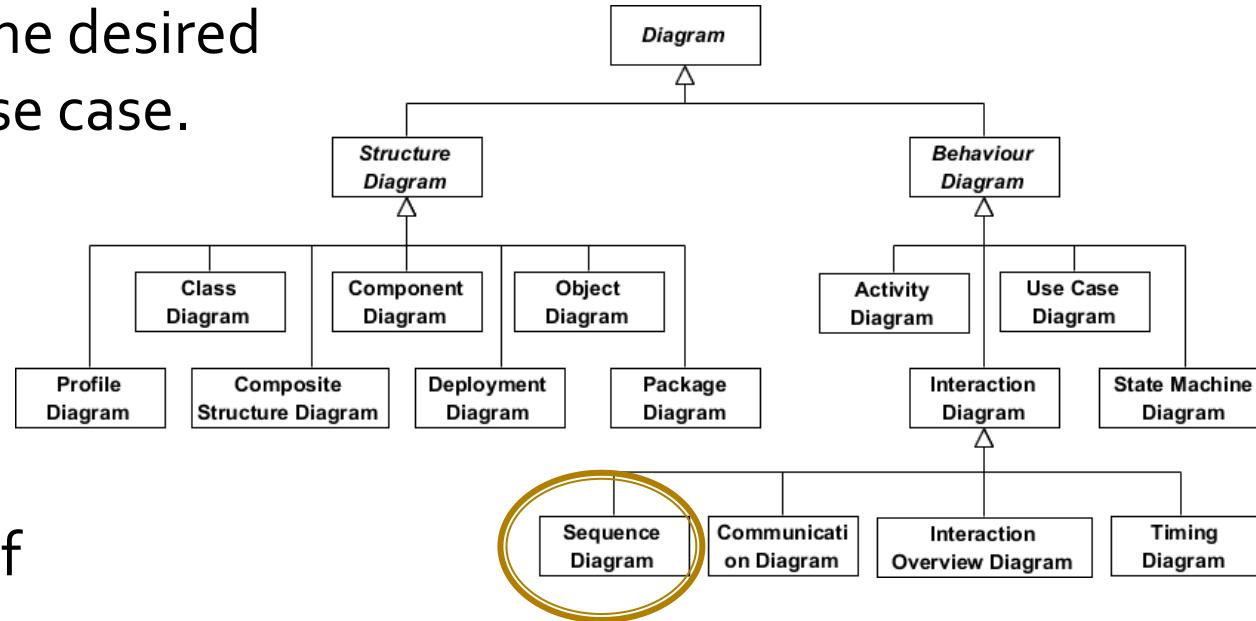
Use Case	User Story
Detailed and well documented. Includes descriptions of interactions between the System and the Actor	Short and succinct in length
Contains extensive coverage of the goal, the trigger(s), preconditions, and flow steps, alternative flow steps, and postconditions that can also act as triggers to another use case	Clear writing structure and often follows this format: a [user] can [action] to achieve [expected outcome]
Often used by developers to develop features based on the descriptions given	Covers a single scenario and can be placed on a story map for project context. Often used to help determine the priority of use cases and which feature flow to implement
Used by the analyst to create a list of feature descriptions and functional requirements for a developer to implement	Used by the analyst to determine user needs in context and determine the priority of features based on business goals, strategies, and user feedback on the currently implemented features
Often used in waterfall software development processes and can be adapted for agile by giving the project a blueprint of direction	Often used in agile software development projects for faster decision making and feature implementation prioritization

序列图 UML Sequence Diagram



Sequence Diagram

- Defines **event sequences** that result in some desired outcome for a single scenario within one use case.
 - The messages sent between objects
 - The order in which messages occur
- Two dimensions
 - Vertical, top down, the **time sequence** of messages/calls as they occur
 - Horizontal, left to right, the **object instances** that the messages are sent to



序列图

Non-persistent HTTP

suppose user enters URL:

`www.someSchool.edu/someDepartment/home.index`

(contains text,
references to 10
jpeg images)

1a. HTTP client initiates TCP

connection to HTTP server
(process) at
`www.someSchool.edu` on port
80

1b. HTTP server at host

`www.someSchool.edu` waiting
for TCP connection at port 80.
“accepts” connection, notifying
client

2. HTTP client sends HTTP *request*

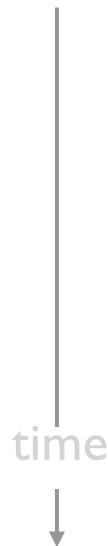
message (containing URL) into
TCP connection socket.
Message indicates that client
wants object
`someDepartment/home.index`

3. HTTP server receives request

message, forms *response*
message containing requested
object, and sends message into
its socket

time

Non-persistent HTTP (cont.)



4. HTTP server closes TCP connection.

5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-5 repeated for each of 10 jpeg objects

Example

A weather data collection system is required to generate weather maps on a regular basis using data collected from remote, unattended weather stations and other data sources such as weather observers, balloons and satellites.

Weather stations transmit their data to the area computer in response to a request from that machine.

The area computer validates the collected data and integrates it with the data from different sources.

The integrated data is archived and, using data from this archive and a digitised map database, a set of local weather maps is created.

Maps may be printed for distribution on a special-purpose map printer or may be displayed in a number of different formats.

Example: Weather System

A weather data collection system is required to generate weather maps on a regular basis using data collected from remote, unattended weather stations and other data sources such as weather observers, balloons and satellites.

Weather stations transmit their data to the area computer in response to a request from that machine.

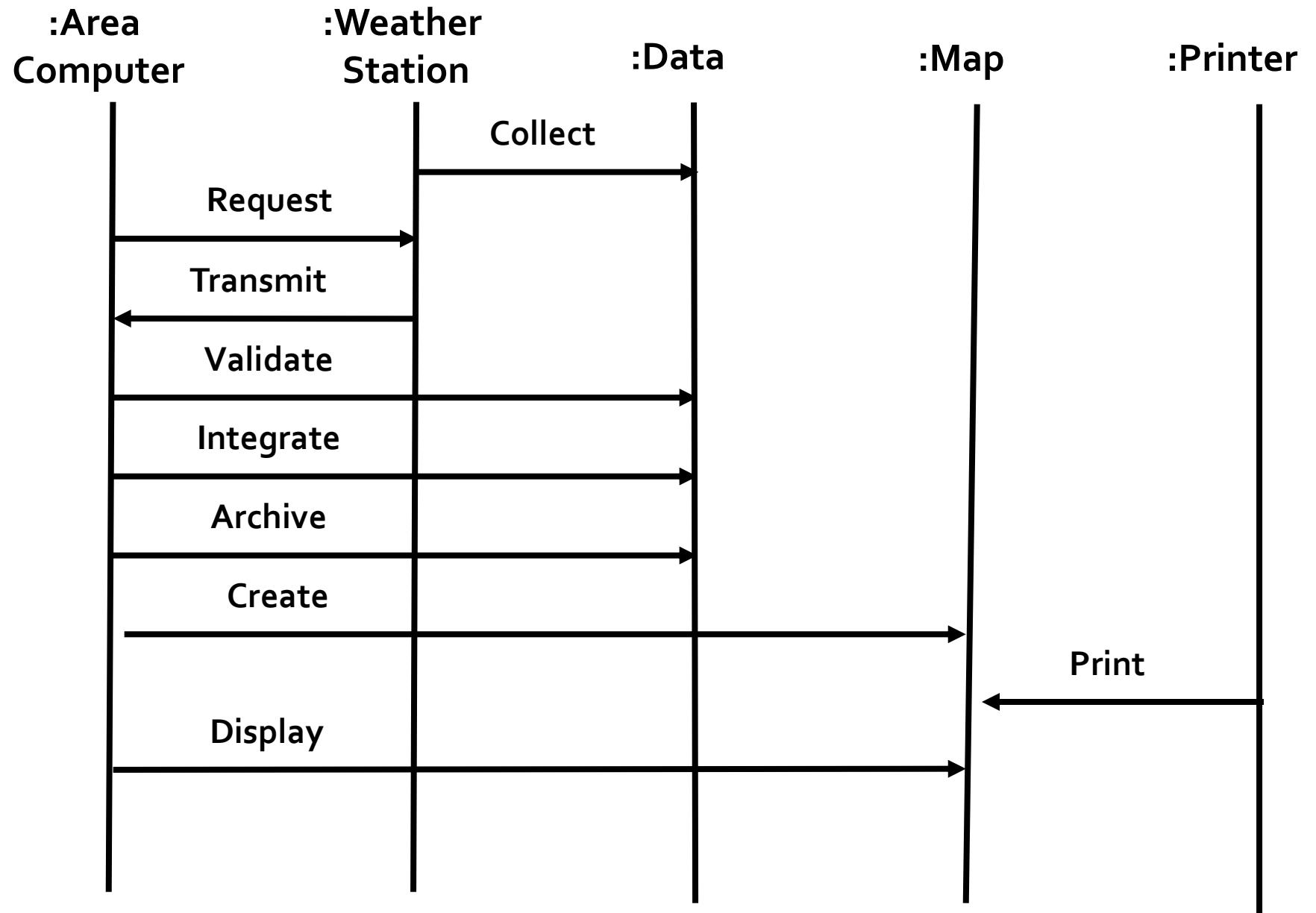
The area computer validates the collected data and integrates it with the data from different sources.

The integrated data is archived and, using data from this archive and a digitised map database a set of local weather maps is created.

Maps may be printed for distribution on a special-purpose map printer or may be displayed in a number of different formats.

Example: Sequence Analysis

- Weather stations *collect* data
- area computer *requests* data *from* Weather stations
- Weather stations *transmit* their data
- area computer *validates* the collected data
- area computer *integrates* data
- area computer *archives* data
- area computer *uses* map database
- area computer *creates* weather maps
- map printer *prints* Maps
- area computer *displays* Maps



Example Weather System Sequence Diagram

内容提要

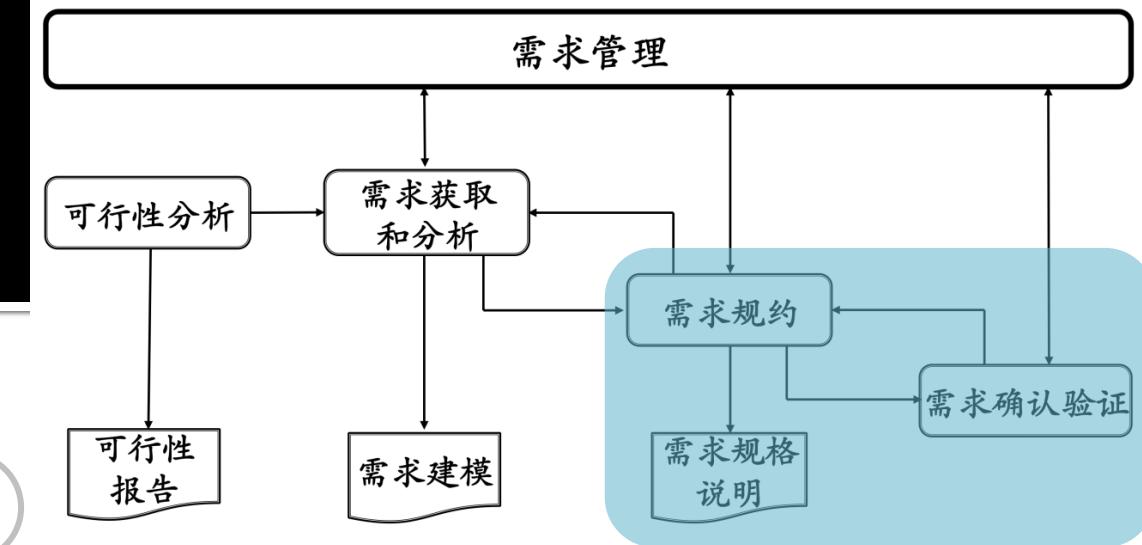
基本概念

需求获取

需求建模

需求规约

需求管理



记录需求 - 软件案例

案例一

软件

非常复杂的流程控制应用系统，参与人数超过100，时间跨度为若干年。

需求记录

高级需求约800个；约20个子项目，每个子项目有各自需求；系统需求引申出大约800~900个需求。

大量的文件...

案例二

软件

中等规模项目。

需求记录

每个项目一个指导文件，包括与项目相关的所有信息：需求，预算、进度计划、质量计划、测试.....

包罗万象的文件...

案例三

软件

大型项目，敏捷开发实践。

需求记录



几个月后，便签条散落一地...

案例四

软件

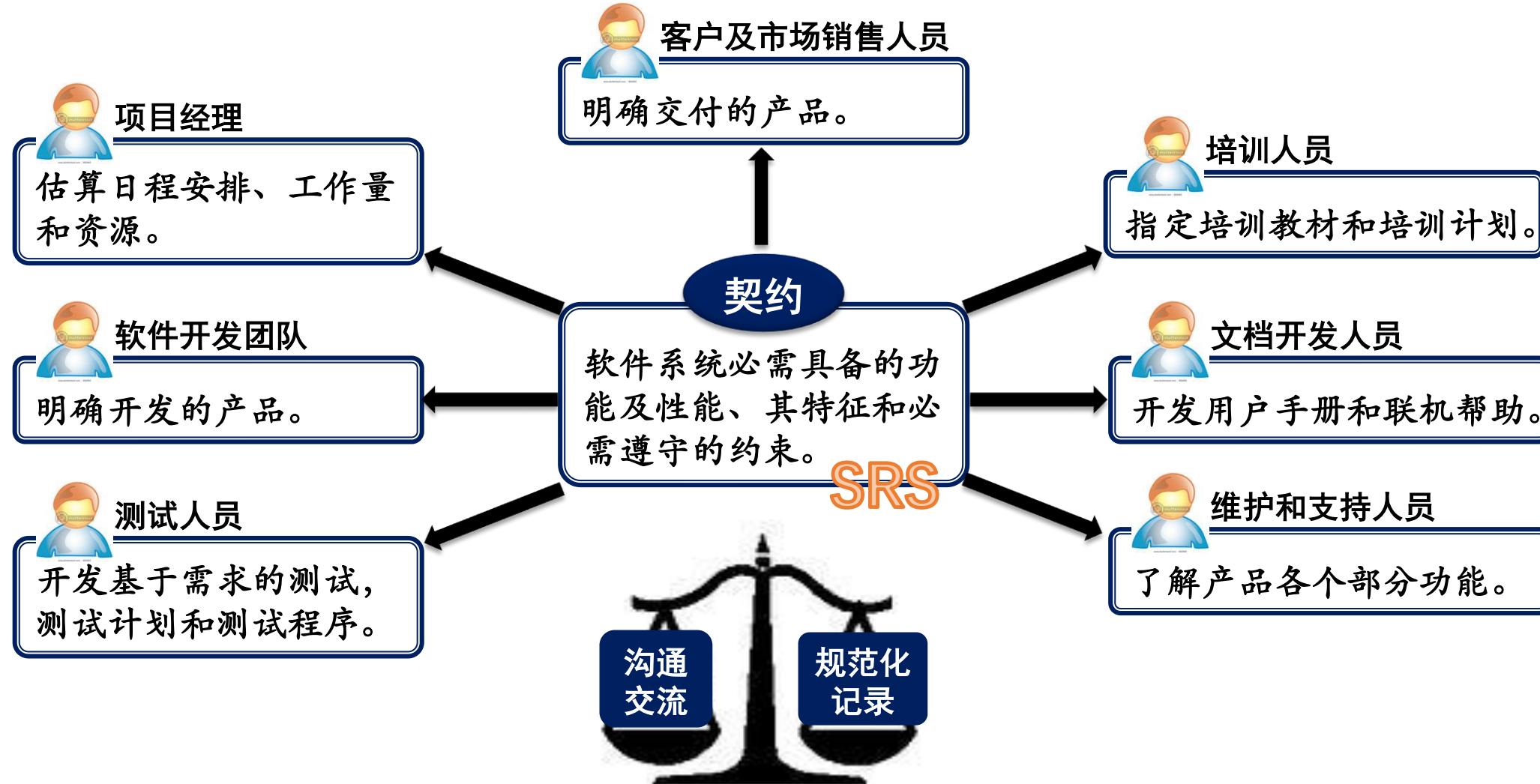
中等项目，需求40-60页。

需求记录

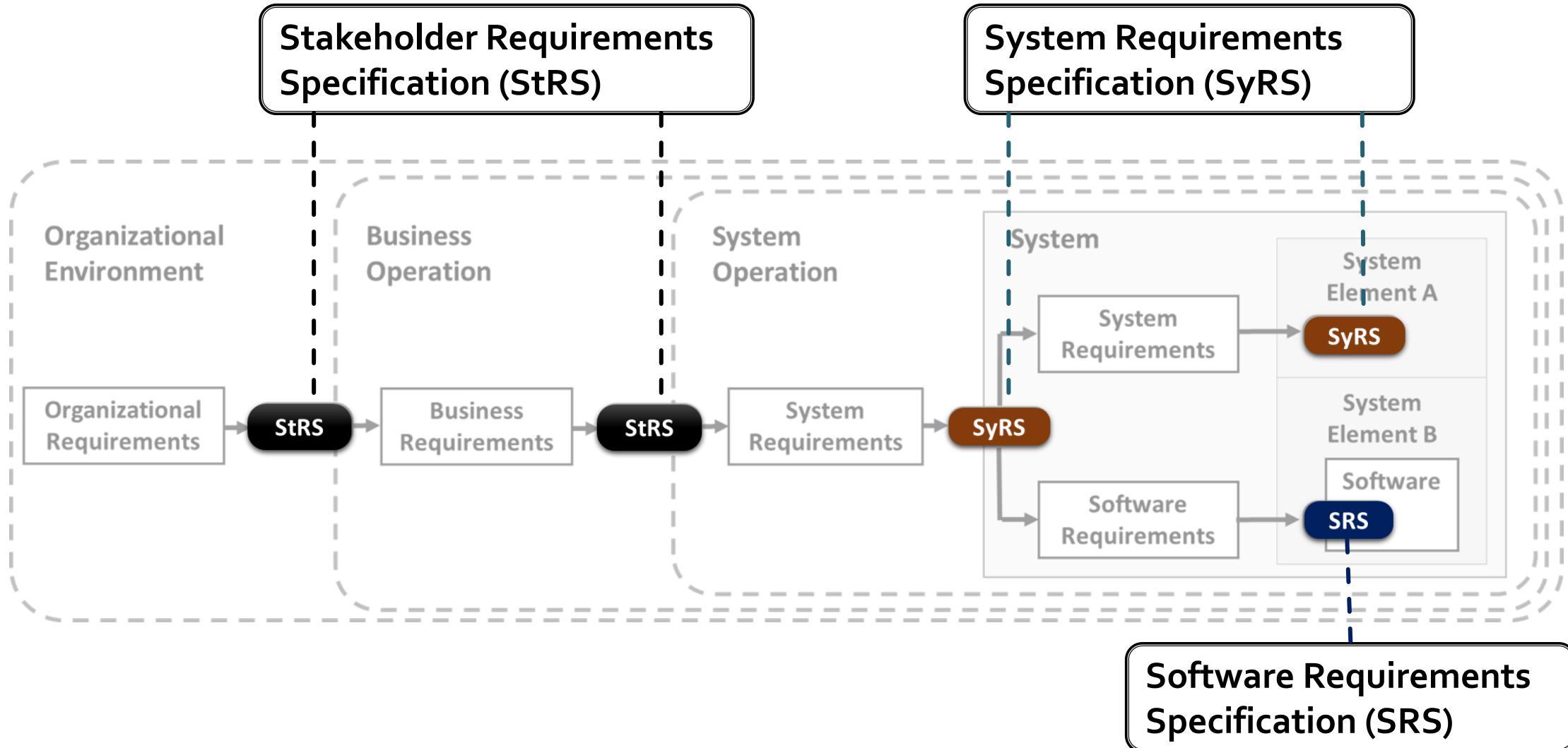
需求说明分为12个独立的文件：一个用于批处理过程，一个用于报表引擎，另一个含有10份报告...

文档爆炸，难以同步变更，难以获取需求全部信息...

软件需求规格说明 SRS



标准：ISO/IEC/IEEE 29148:2011



Stakeholder Requirements Specification (StRS)

Business purpose
Business scope
Business overview
Stakeholders
Business environment
Goal and objective
Business model
Information environment
Business processes
Business operational policies and rules
Business operational constraints
Business operation modes
Business operational quality
Business structure
User requirements
Operational concepts
Operational scenarios
Project constraints

System Requirements Specification (SyRS)

System purpose
System scope
System overview
Functional requirements
Usability requirements
Performance requirements
System interfaces
System operations
System modes and states
Physical characteristics
Environmental conditions
System security
Information management
Policies and regulations
System life cycle sustainment
Packaging, handling, shipping and transportation
Verification
Assumptions and dependencies

Software Requirements Specification (SRS)

Purpose
Scope
Product perspective
Product functions
User characteristics
Limitations
Assumptions and dependencies
Apportioning of requirements
Specific requirements
External interfaces
Functions
Usability requirements
Performance requirements
Logical database requirements
Design constraints
Standards compliance
Software system attributes
Verification
Supporting information

需求分层与分类

横看成岭侧成峰
远近高低各不同



业务需求

业务目标

航空公司业务系统：
把机场的柜台工作人
员成本降低 25%

希望获得的业务收
益，提出系统需求
的客户的业务目标



愿景和范围
文档

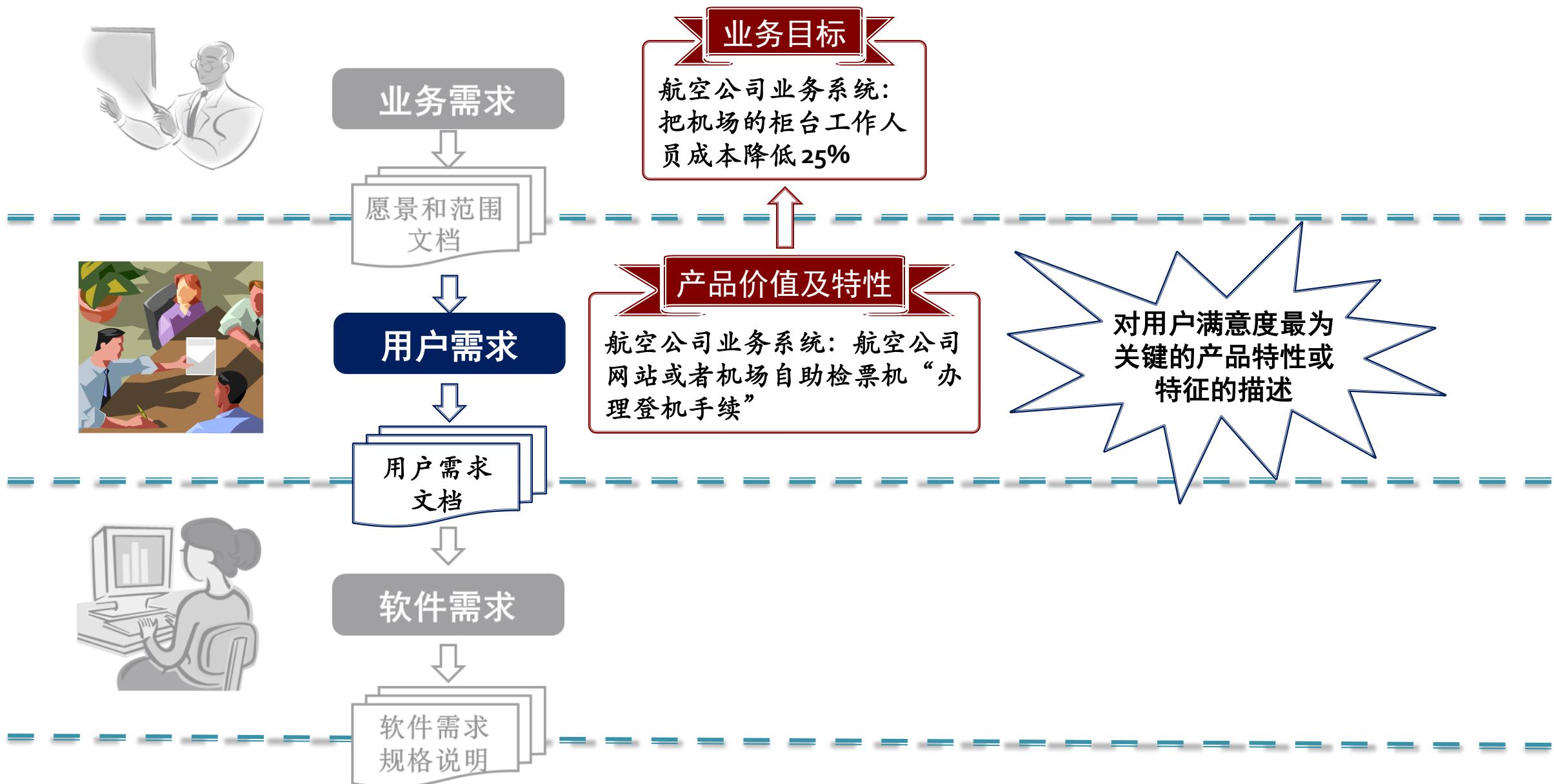
用户需求

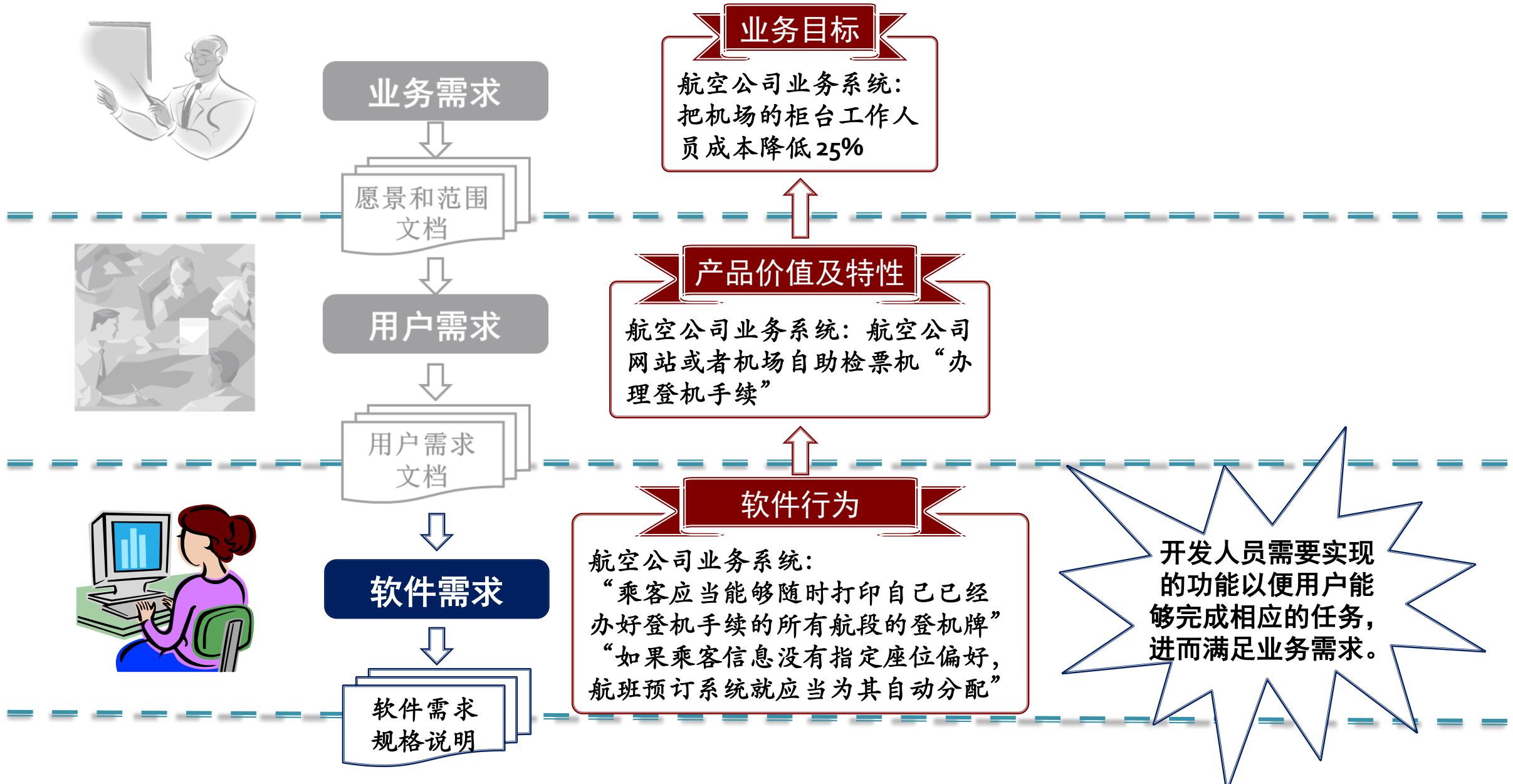
用户需求
文档

软件需求

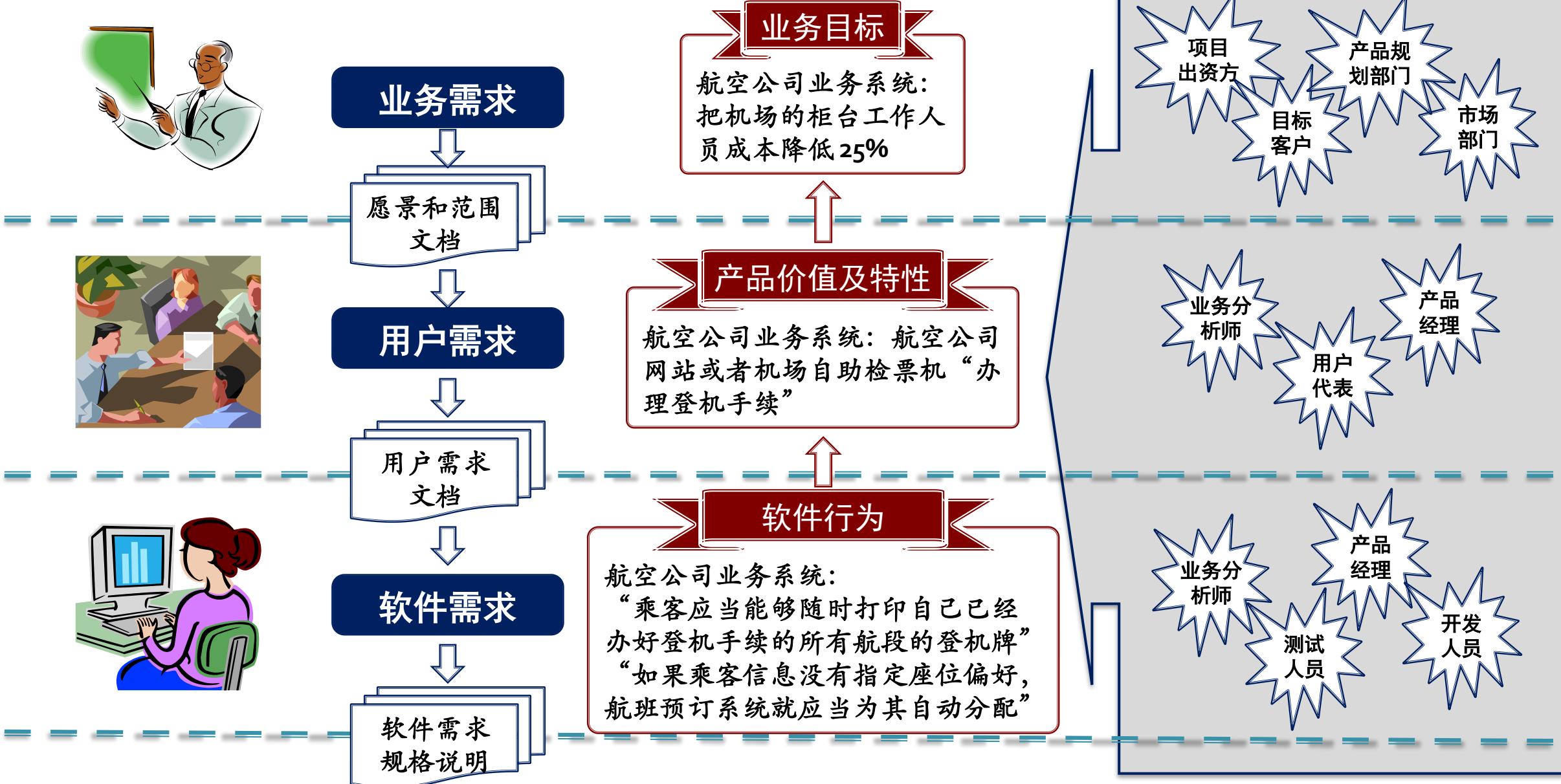
软件需求
规格说明



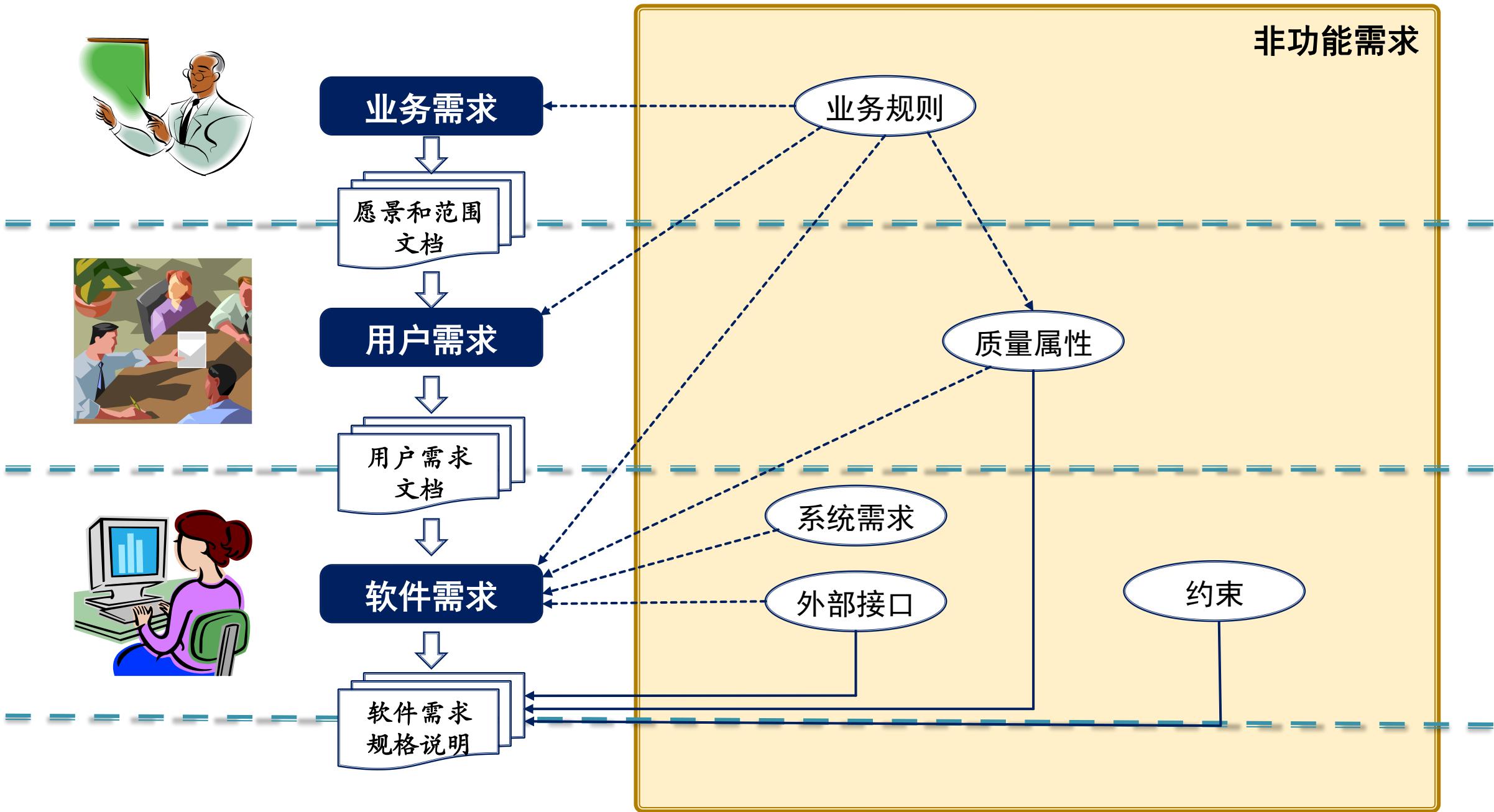




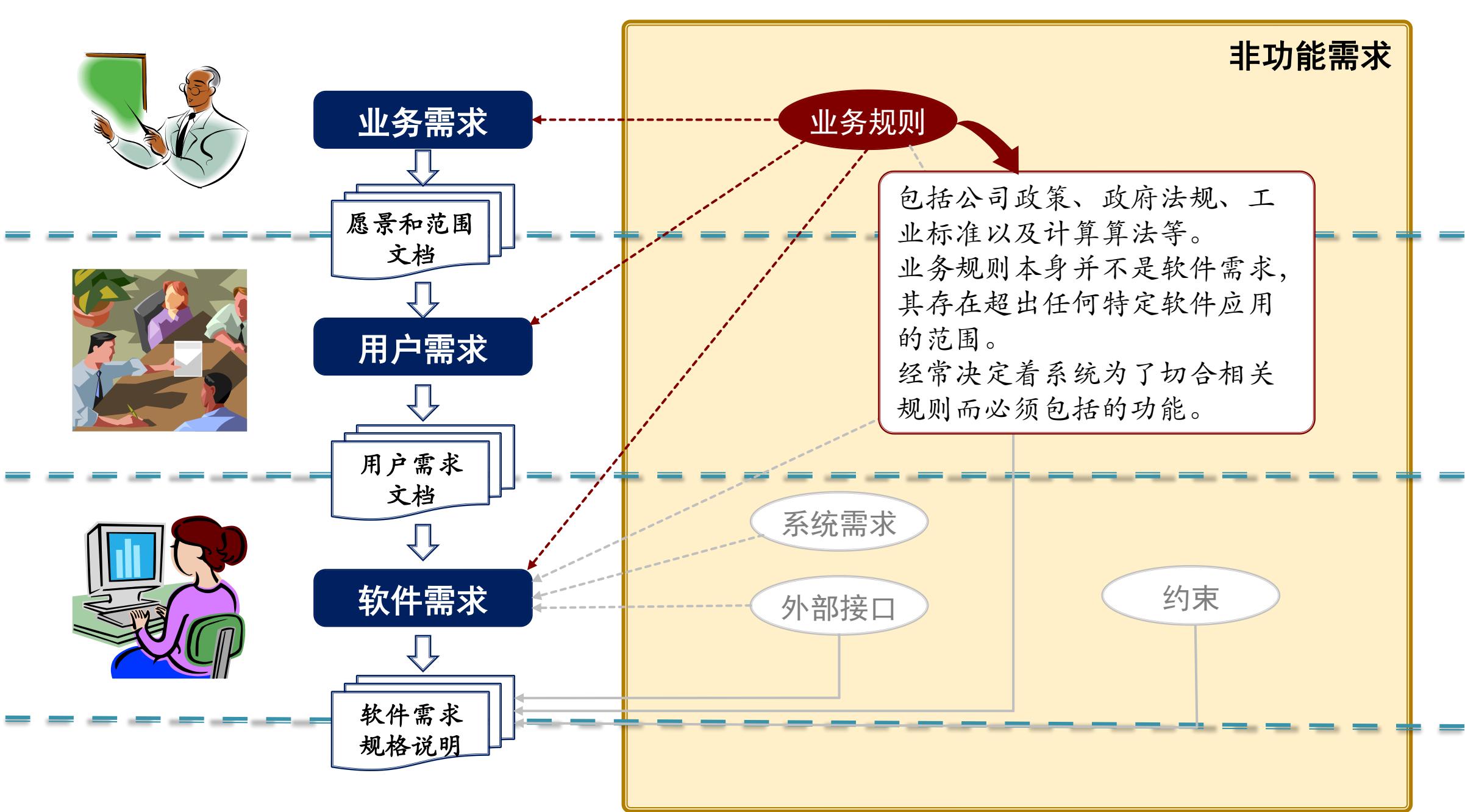
共利益者



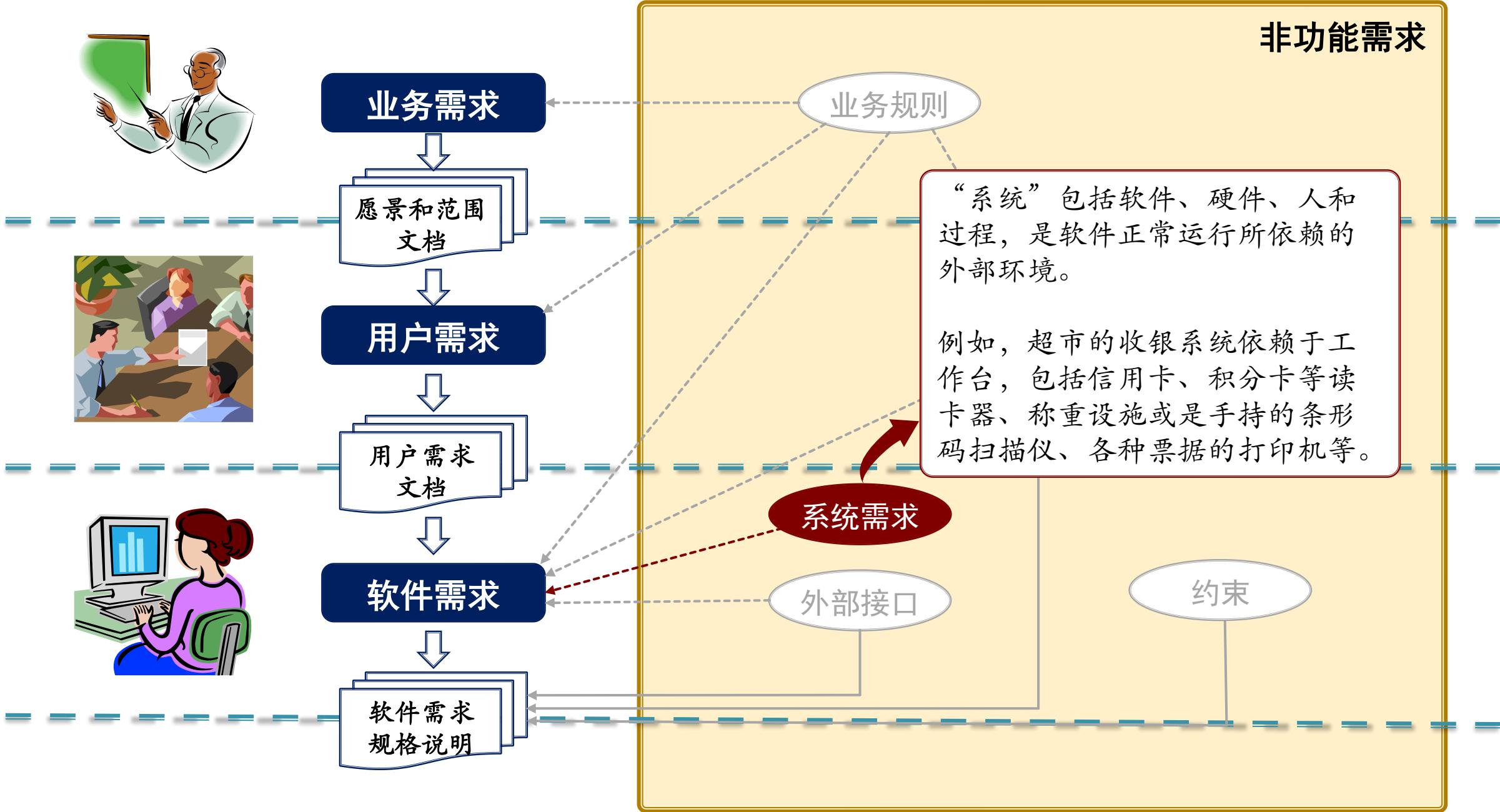
非功能需求



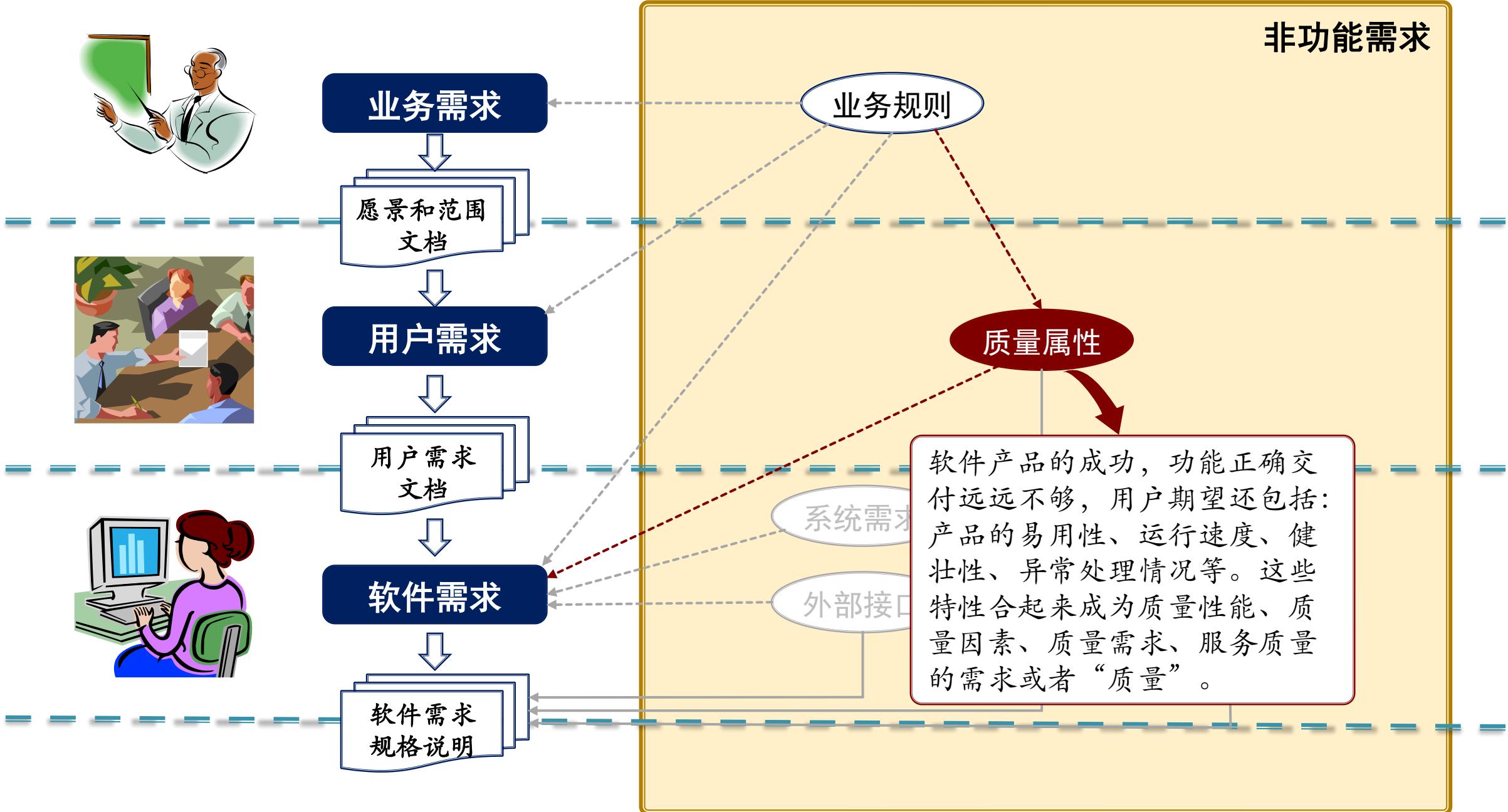
非功能需求



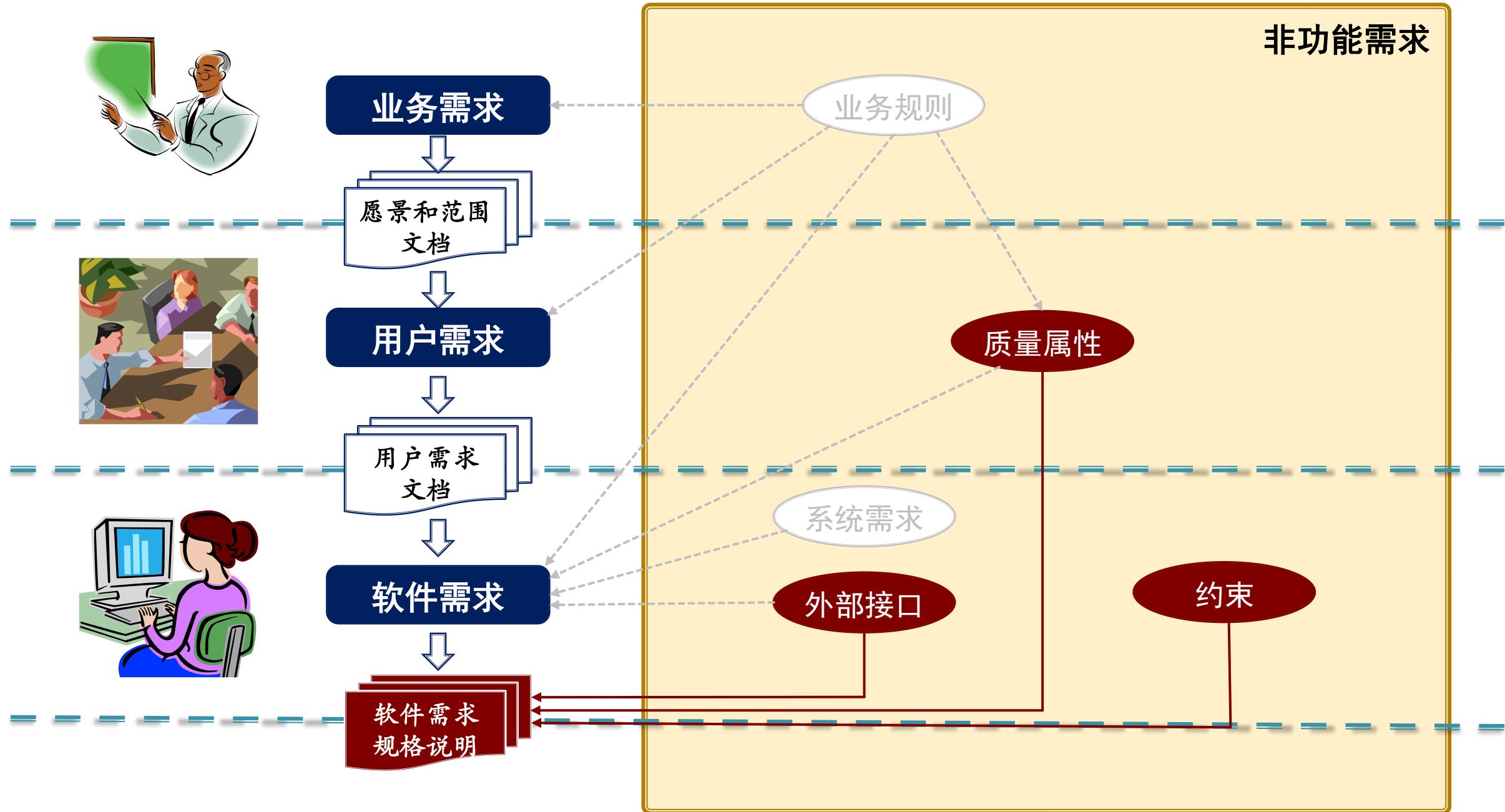
非功能需求



非功能需求



非功能需求



Well-Formed Requirement

- can be verified,
- has to be met or possessed by a system to solve a stakeholder problem or to achieve a stakeholder objective,
- is qualified by measurable conditions and bounded by constraints, and
- defines the performance of the system when used by a specific stakeholder or the corresponding capability of the system, but not a capability of the user, operator, or other stakeholder.

[Condition]
[条件]

[Subject]
[主体]

[Action]
[动作]

[Object]
[客体]

[Constraints]
[约束]

[Value]
[值]

需求样例

[Condition]

[Subject]

[Action]

[Object]

[Constraints]

[Value]

The system [Subject] shall set [Action] the signal x [Object].

When signal x is received [Condition], the system [Subject] shall set [Action] the signal x received [Object] within [Constraints] 2 seconds [Value].

The Radar system [Subject] shall detect [Action] targets [Object].

At sea state 1 [Condition], the Radar system [Subject] shall detect [Action] targets [Object] at ranges out to [Constraints] 100 nautical miles [Value].

The Invoice system [Subject] shall display [Action] invoices [Object].

The Invoice system [Subject] shall display [Action] pending customer [Constraints] invoices [Object] in ascending order [Value] in which invoices are to be paid [Constraints].

Constraints 约束限制

- Constraints restrict the design solution or implementation of the system engineering process.
- May apply across all requirements
- May be specified in a relationship to a specific requirement or set of requirements
- May be identified as stand-alone requirements

When signal x is received [Condition], the system [Subject] shall set [Action] the signal x received [Object] within [Constraints] 2 seconds [Value].

两个动作（接收信号 receive，设置信号位 set）之间的约束：时间延迟

At sea state 1 [Condition], the Radar system [Subject] shall detect [Action] targets [Object] at ranges out to [Constraints] 100 nautical miles [Value].

动作 detect 的约束：作用范围

The Invoice system [Subject] shall display [Action] pending customer [Constraints] invoices [Object] in ascending order [Value] in which invoices are to be paid [Constraints].

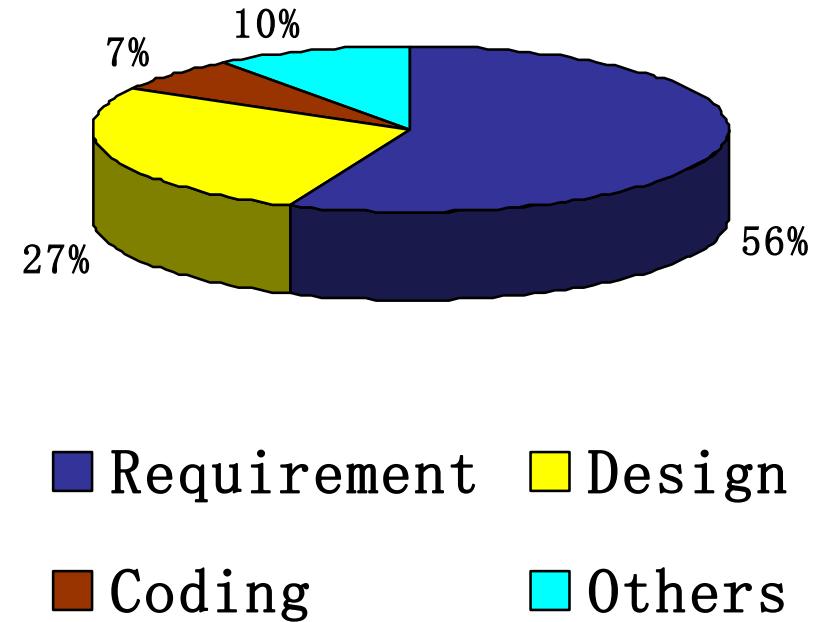
动作 display 的约束：作用范围（待付款客户，将付款的发票）和形式（升序）

需求改写

- 美团外卖送餐要快
- 共享单车扫码开车要快

需求中的缺陷

“ Requirements..... are *major source of expensive bugs*. The range is from a few percent to more than 50%, depending on application and environment. What hurts most about these bugs is that they're *the earliest to invade the system and the last to leave*. It's not unusual for a faulty requirement to get through all development testing, beta testing, and initial field use, only to be caught after hundreds of sites have been installed.”



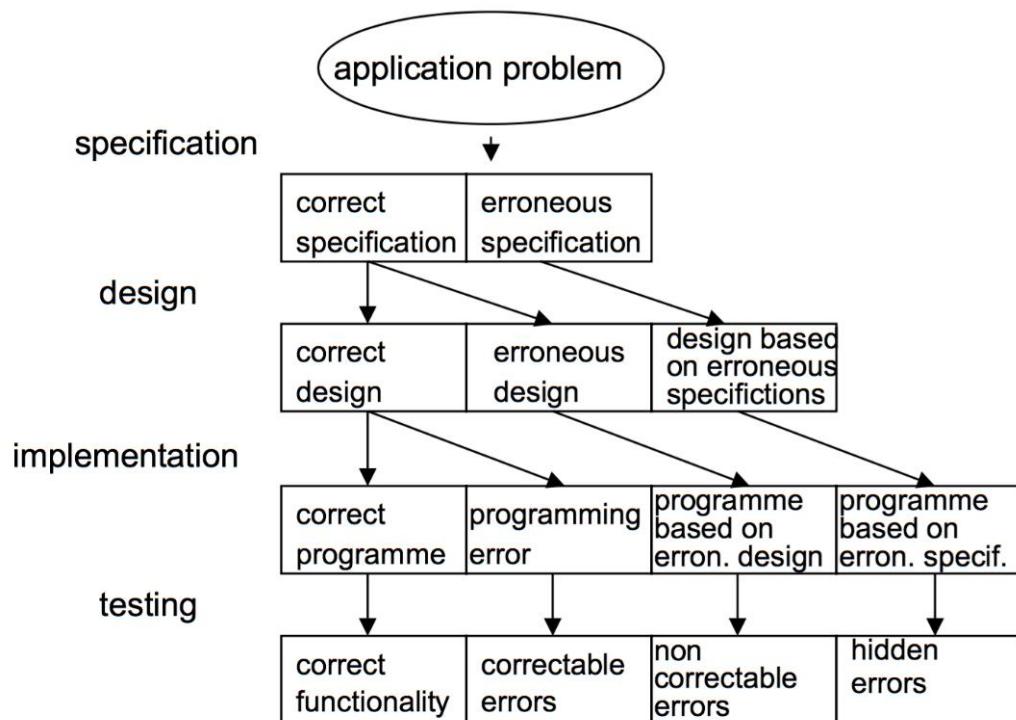
— Beizer, 1990

需求中缺陷的代价

Davis' Hypotheses regarding the Importance of Requirements Specifications ([Davis])

The later in the life-cycle that a fault will be discovered,
the more expensive it will be to fix it .

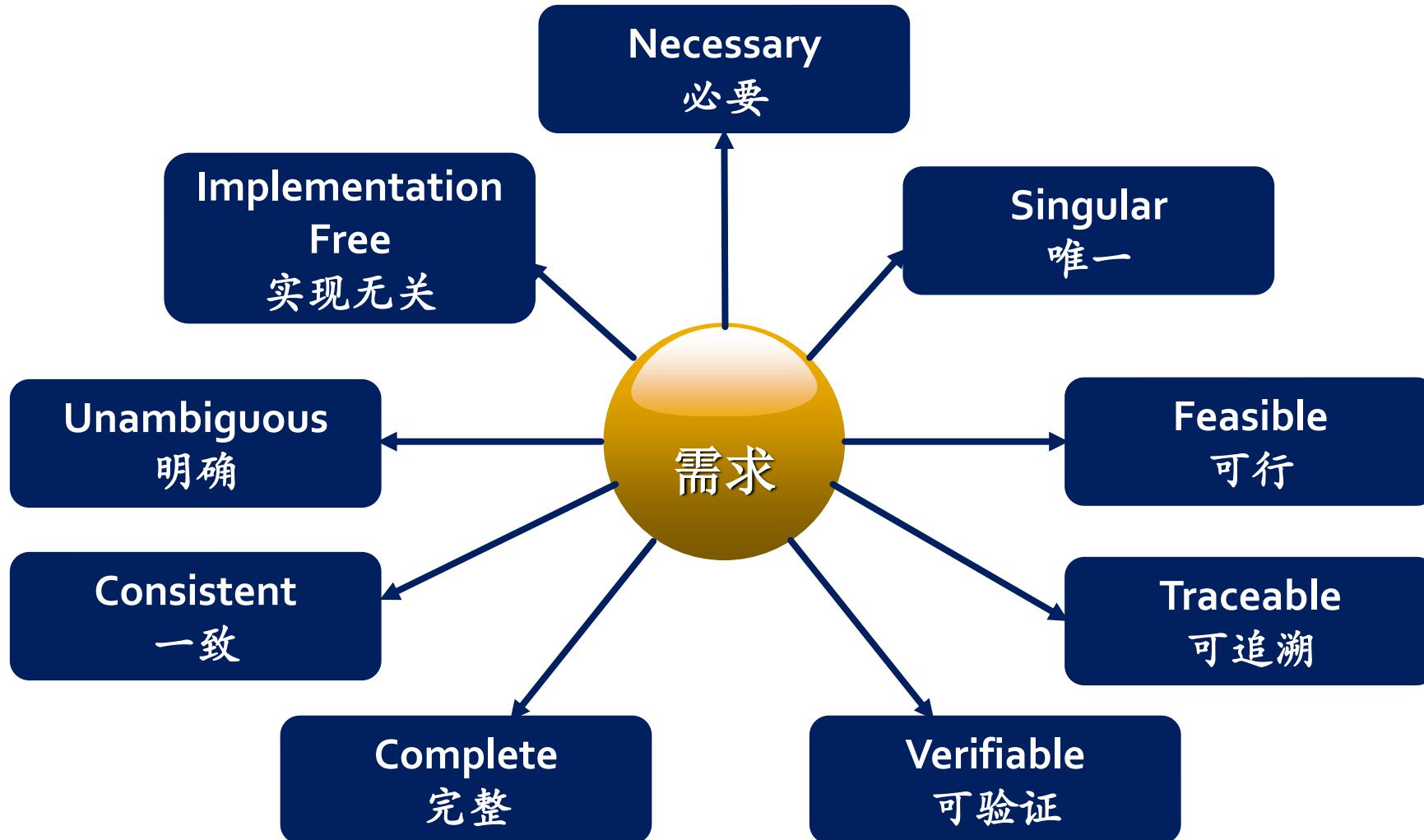
- ▶ Mizuno's observation:



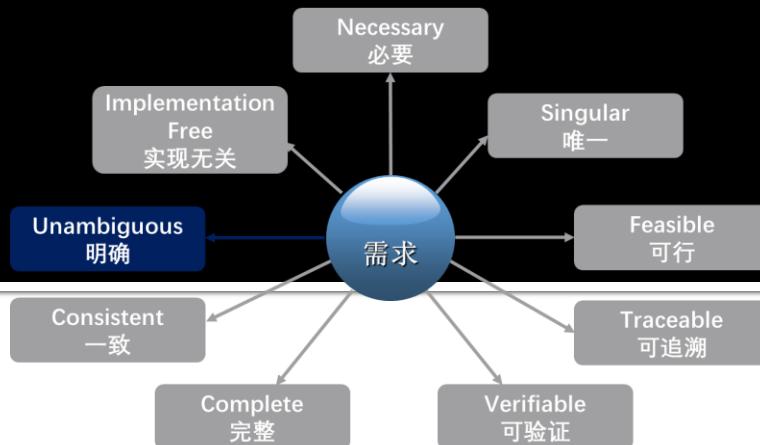
需求缺陷发现的时间	相对代价
需求分析	0.1 ~ 0.2
设计	0.5
编码	1
单元测试	2
系统测试	5
软件维护	20

需求质量特性

ISO/IEC/IEEE 29148:2011

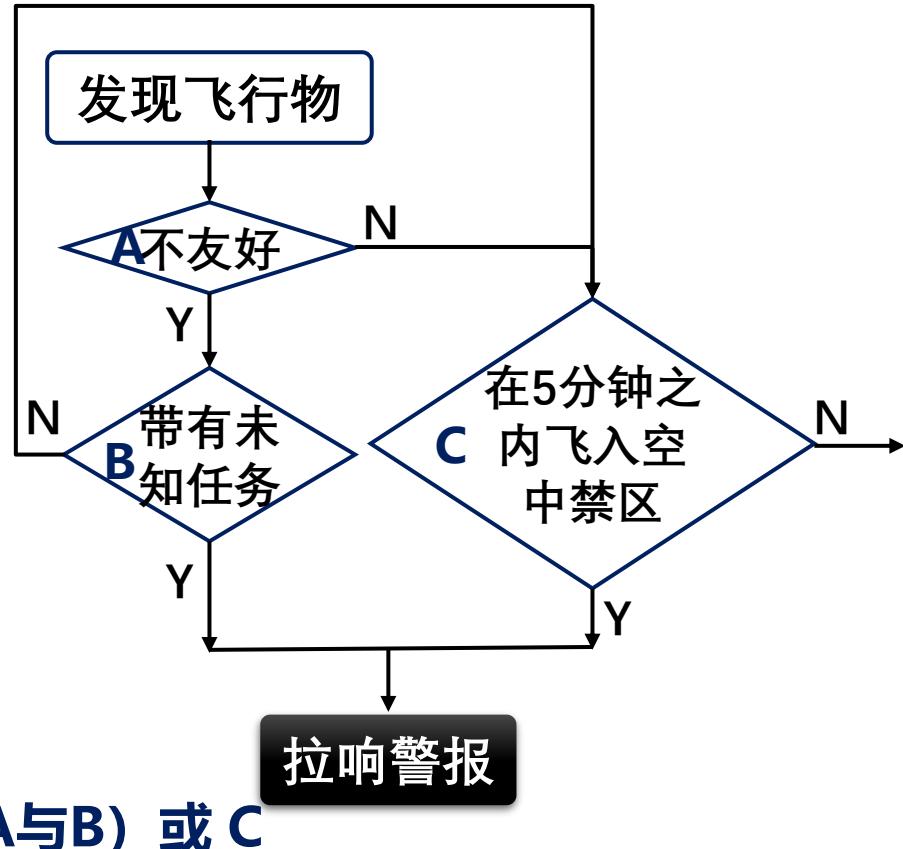


明确

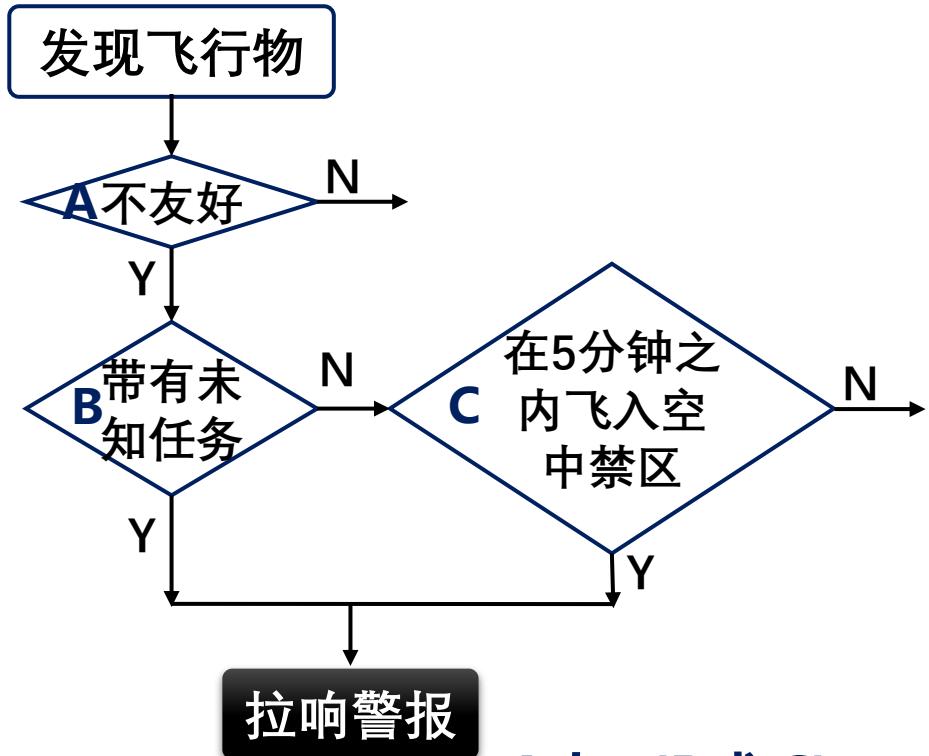


■ 正确、准确、无二义性

“发现任何不友好并且带有未知任务的或者有可能在5分钟之内飞入空中禁区的飞行物时要拉响警报”



(A与B) 或 C



A 与 (B或 C)

明确

- 需求中应避免使用的歧义性词语
 - 尽可能、最好的、最多的：明确**量值**
 - 不少于、不超过、在...之间：明确**边界值**
 - 有效地、迅速的、灵活的：明确**标准**
 - 通常、大多数情况下、理想状况下：说明**异常情况**
 - 支持...功能：说明支持的**内涵**
 - 匹配的、符合的、同样的：明确比较的**标准**



可验证性

可以基于客观的方法，检验需求是否被正确实现。

不完整、不一致、不可实现或是有歧义的需求都是不可验证的。

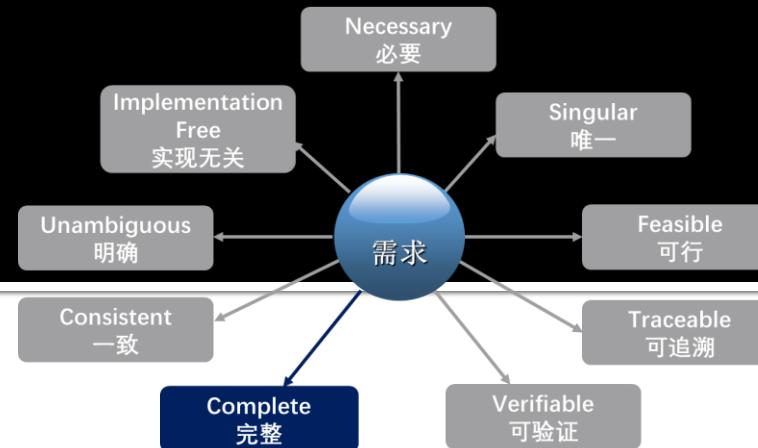
完整

- 每个需求包括了必要的信息
- 需求文档覆盖所有**功能和约束**

如果没有选择最优计划也不提供证据，客户应该自动选择默认的基本计划。



- 选择了最优计划，没有提供证据。
- 选择了最优计划，提供了证据。
- 没有选择最优计划，提供了证据。



条件	1	2	3	4
选择最优计划	Y	Y	N	N
提供证据	Y	N	Y	N
动作				
自动选择默认的基本计划			?	X

需求改进练习

系统必需检查活动账户日志和账户管理存档之间账户数据的一致性。用来产生比较数据的逻辑应该基于已有一致性检查工具的逻辑。换句话说，新代码不需要从零开始开发。然而，需要加入额外的逻辑来识别数据库是否是一个权威的信息来源。新功能将包括往表里写数据来表示怎样/在哪里解决不一致性。另外，代码还要检查违反安全工具数据库的异常情形。一旦违反规定，自动警告邮件应该发送给安全法规团队。

- ① “已有”的工具是什么工具？如何“基于”已有工具？即和已有工具的接口及集成方式？
- ② “新代码”如何基于已有代码开发？已有代码如何重用？如何扩展？
- ③ “额外的逻辑”都有哪些？
- ④ 如何定义信息来源的权威性？如何识别？
- ⑤ “新功能”还包括哪些？
- ⑥ “怎样/在哪里”需要明确具体是什么意思。
- ⑦ 这段文字中，包含了大量的信息，需要拆分为独立的需求。

内容提要

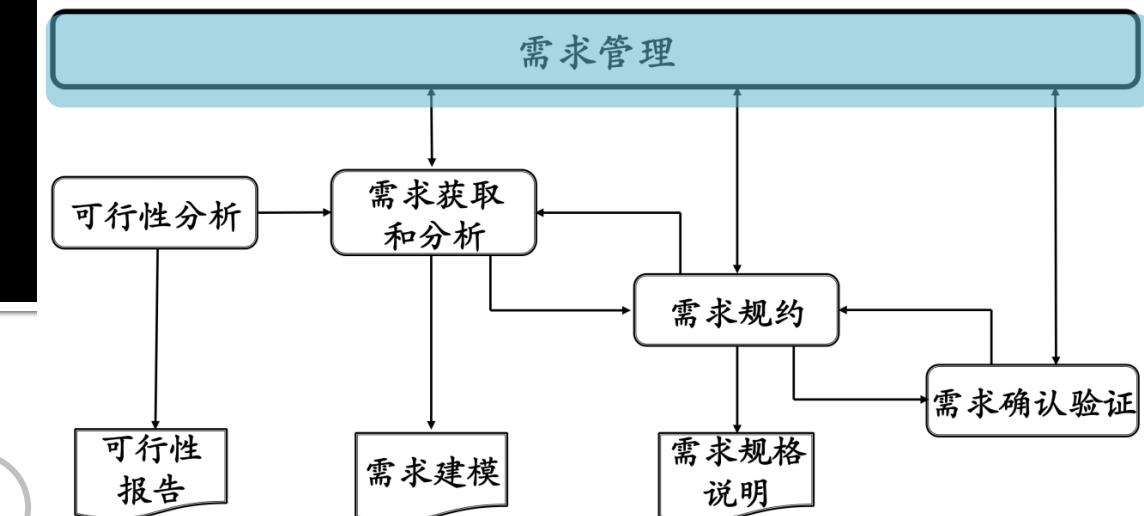
基本概念

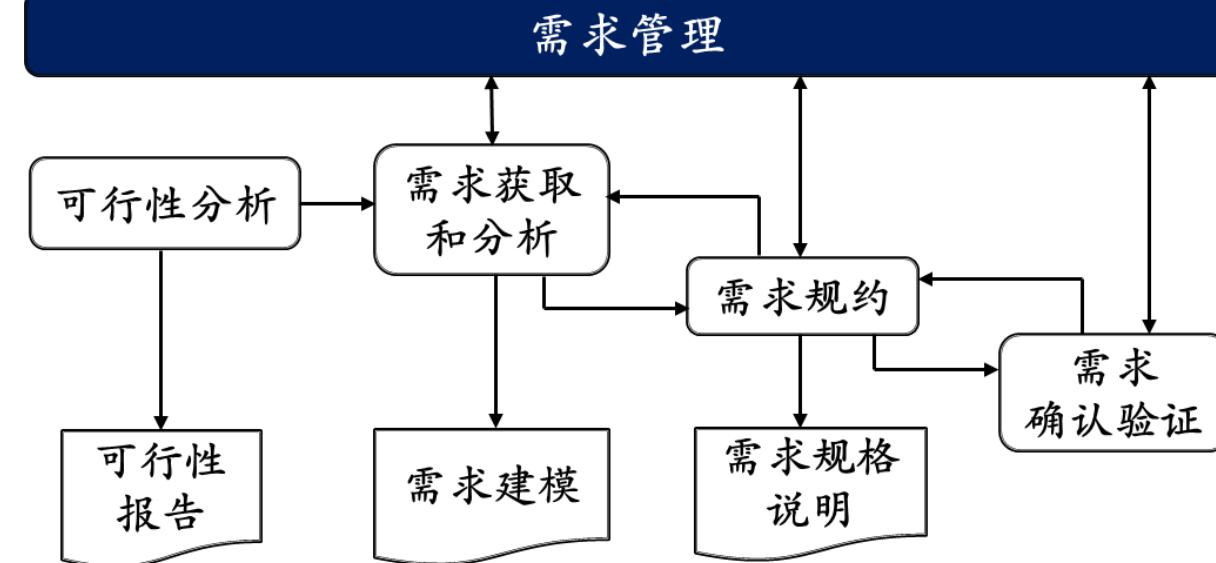
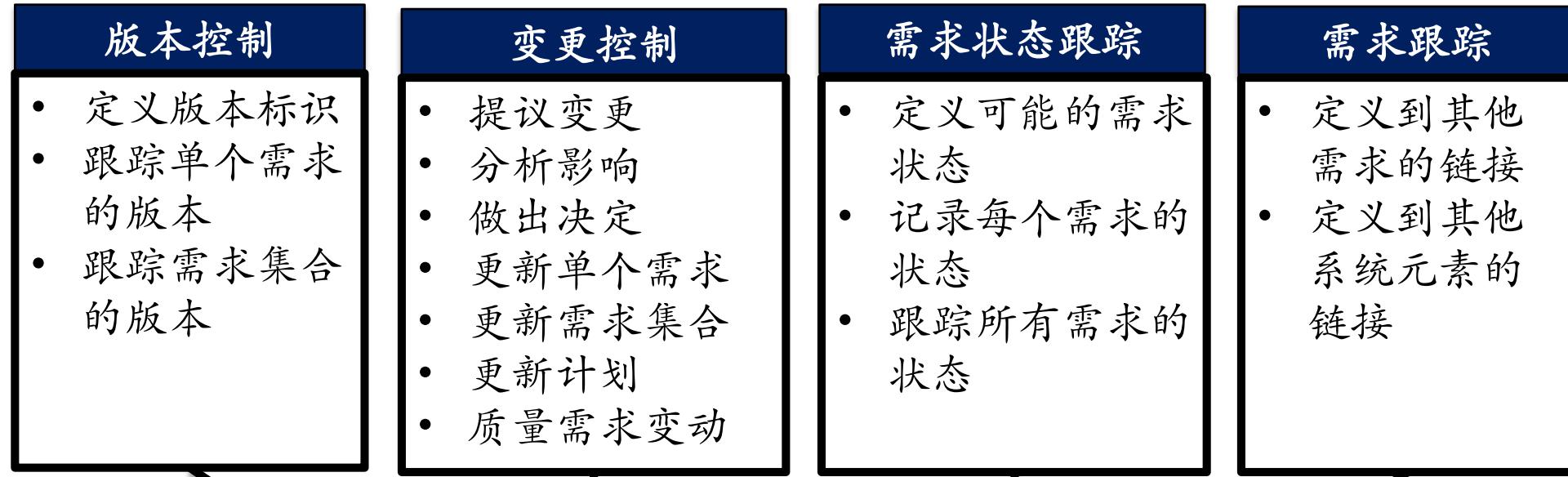
需求获取

需求建模

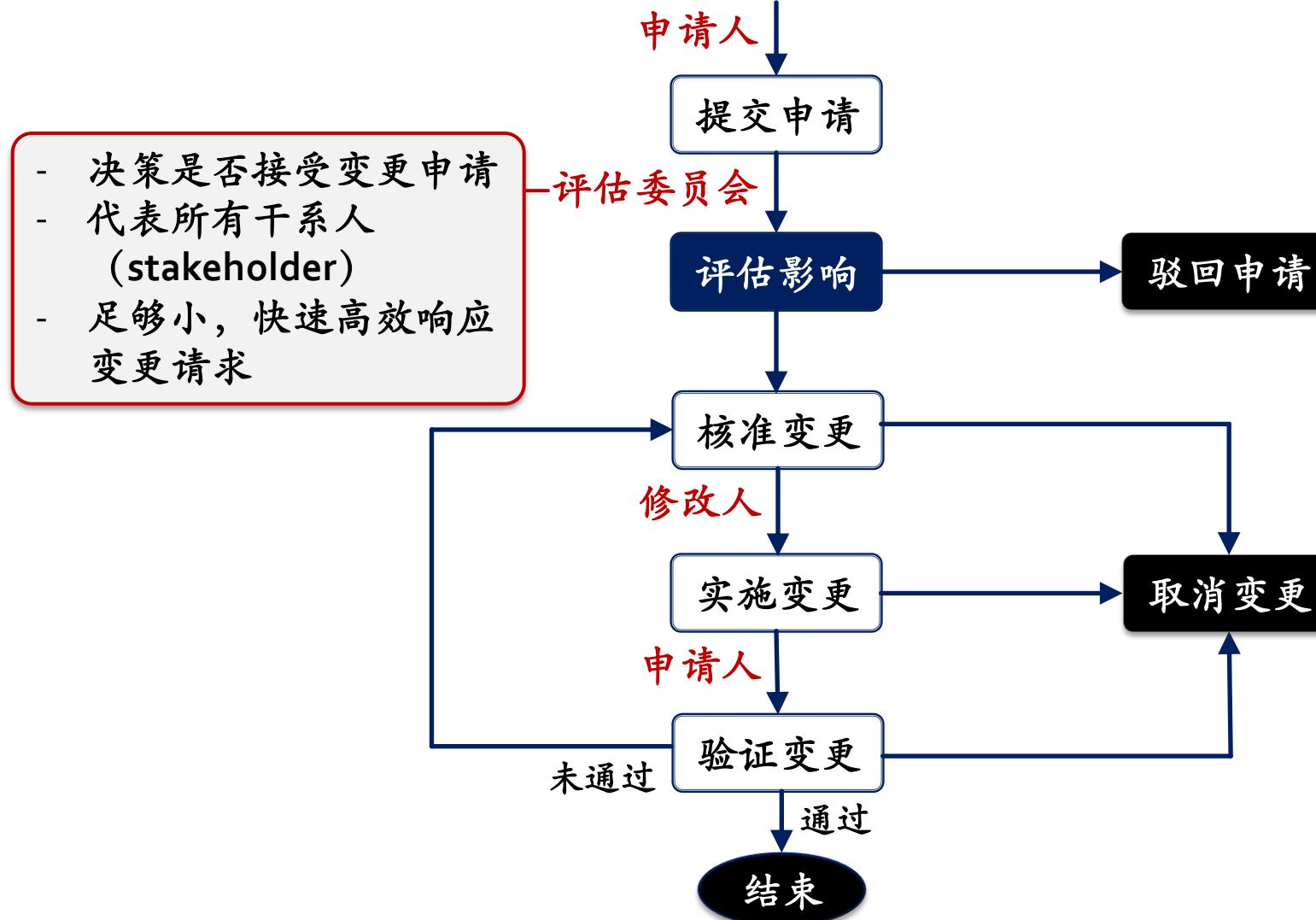
需求规约

需求管理





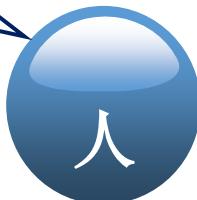
变更控制流程



变更影响分析

所有受到影响的人员都应该知晓、确认变更。

- 用户：业务决策，变更必要性，具有价值
- 开发团队：承诺完成
- 管理人员：调配资源
-

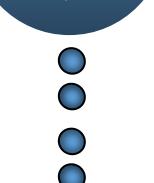


变更可能造成

- 工期延误
- 增加人员
- 额外的软、硬件投入
-



变更



时

针对项目的各个环节，
评估变更可行性和代价

- 需求分析
- 软件设计
- 编码
- 测试
- 项目管理
- ...

任务	工时
更新SRS或需求库	
创建新的设计组件	
修改已有的设计组件	
开发新的用户接口组件	
修改已有的用户接口组件	
开发新的用户文档和帮助界面	
开发新的源代码	
修改已有的源代码	
获取授权并集成第三方软件	
开发新的单元测试和集成测试	
修改已有的单元测试和集成测试	
.....	
总估算时间	

参考文献

- 《软件需求》（第三版），[美] Karl Wiegers, Joy Beatty. 清华大学出版社，2016年3月。
- 《软件工程》（第十版），[英] Ian Sommerville. 北京机械工业出版社，2018年1月。
- 《软件工程：实践者的研究方法》（第七版），[美] R. S. Pressman. 北京机械工业出版社，2010年9月。
- Charette RN. Why software fails. IEEE spectrum. 2005 Sep;42(9):36.
- "The chaos report." *The Standish Group* (2014).

Thank you!

