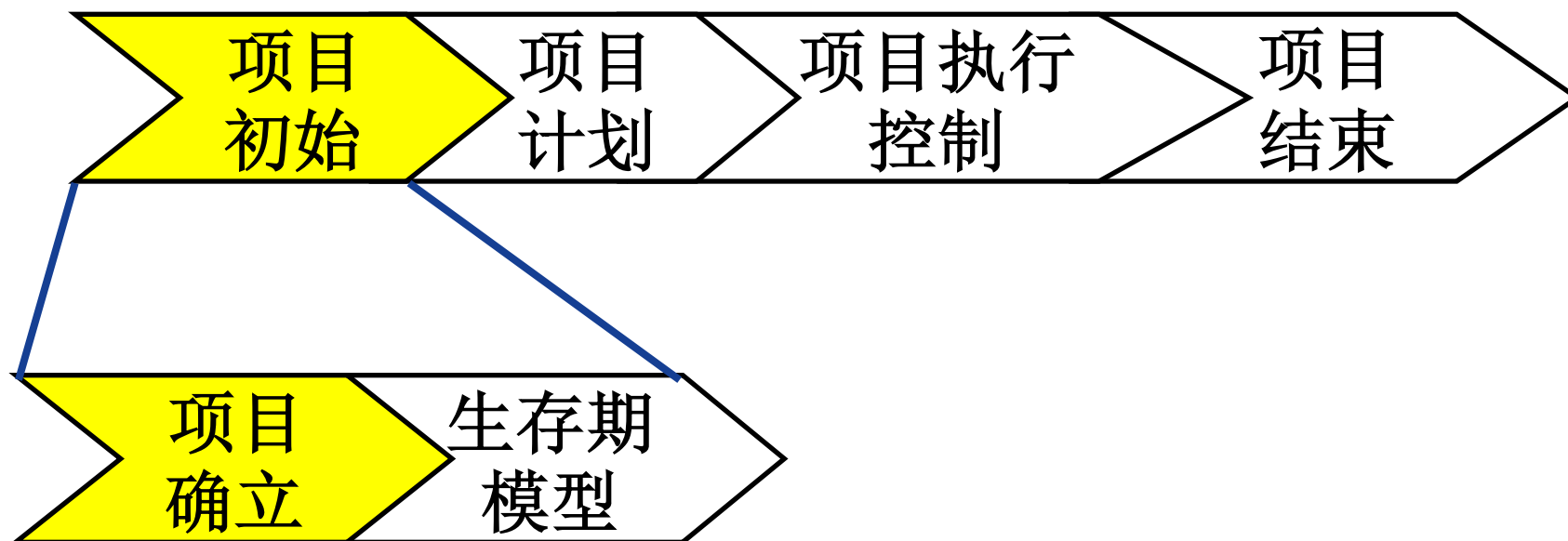


# 软件项目管理的阶段

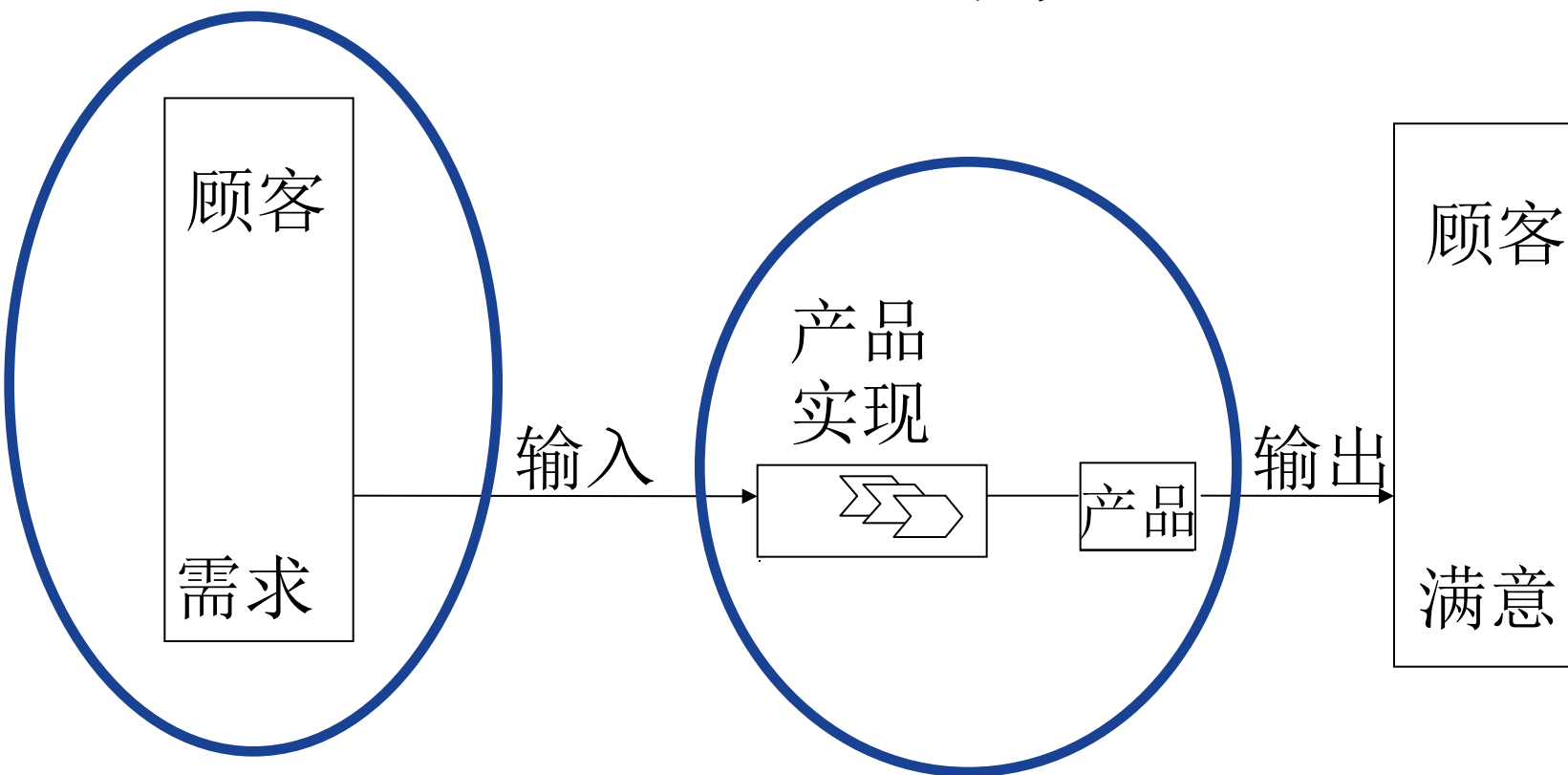


# 软件项目管理

## 第一篇 项目初始

# 软件项目启动

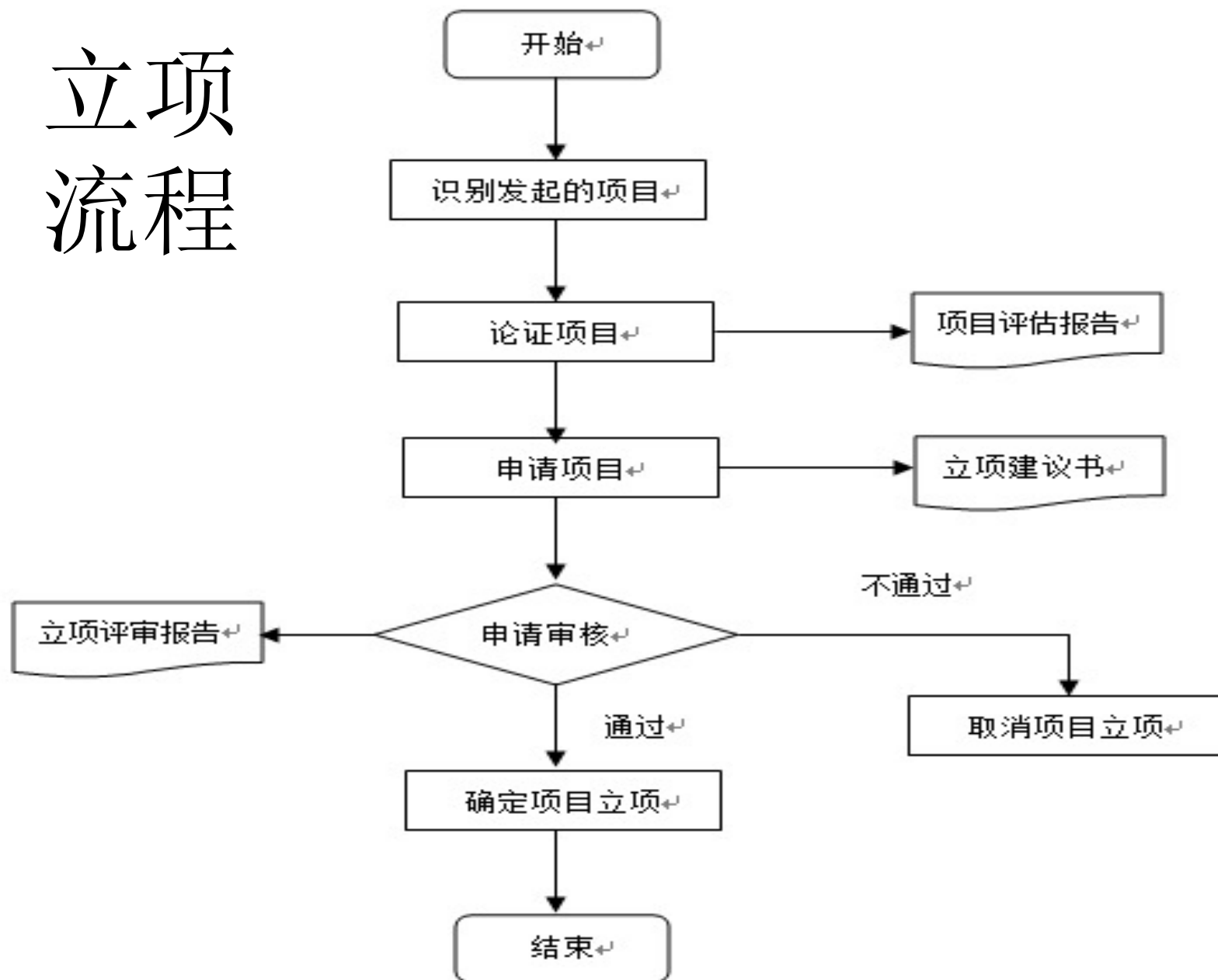
甲方 ↔ 乙方



- ❑ 一、项目立项
- ❑ 二、授权项目
- ❑ 三、初始项目分析
- ❑ 四、生存期模型
- ❑ 五、案例分析

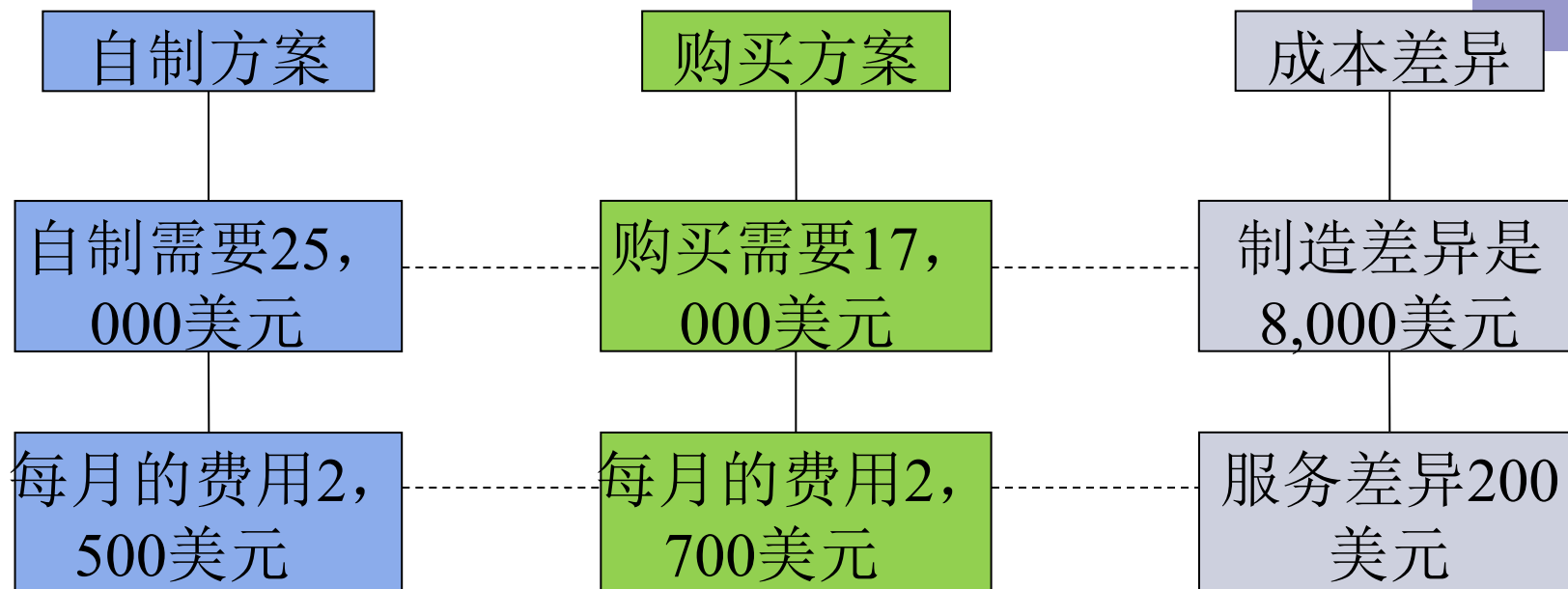
明确项目的目标、时间表、项目使用的资源和经费，而且得到执行该项目的项目经理和项目发起人的认可。这个阶段称为立项阶段。

# 立项流程



# 自造-购买决策 Make or Buy

MIMA



解决方案

自制方案可以承受的月份数: $8000/200=40$

如果软件的生存期在40个月以内,可以选择购买方案.

如果软件的生存期不在40个月以内,可以选择自制方案.

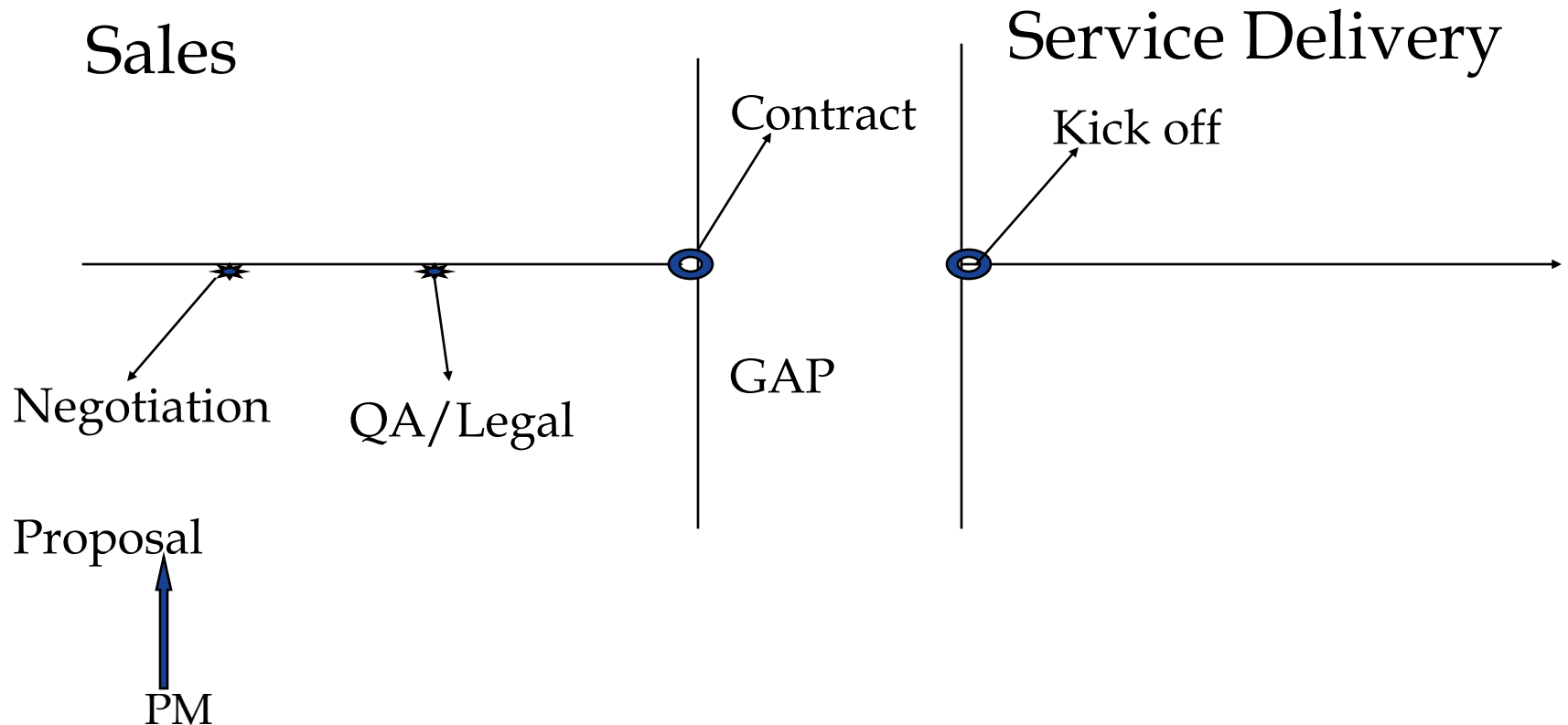
# 项目立项

---

MIMA

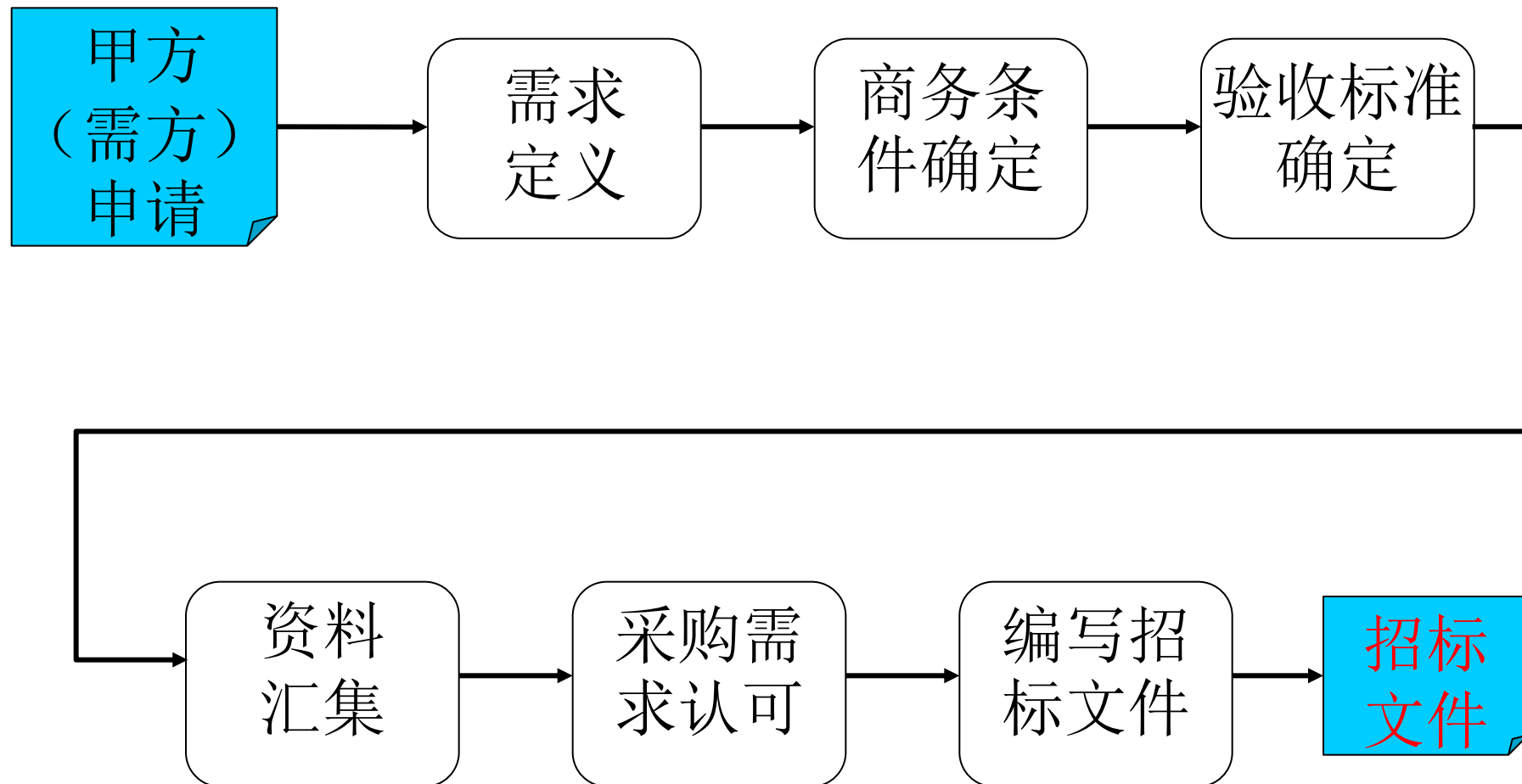
- 合同项目
- 内部项目



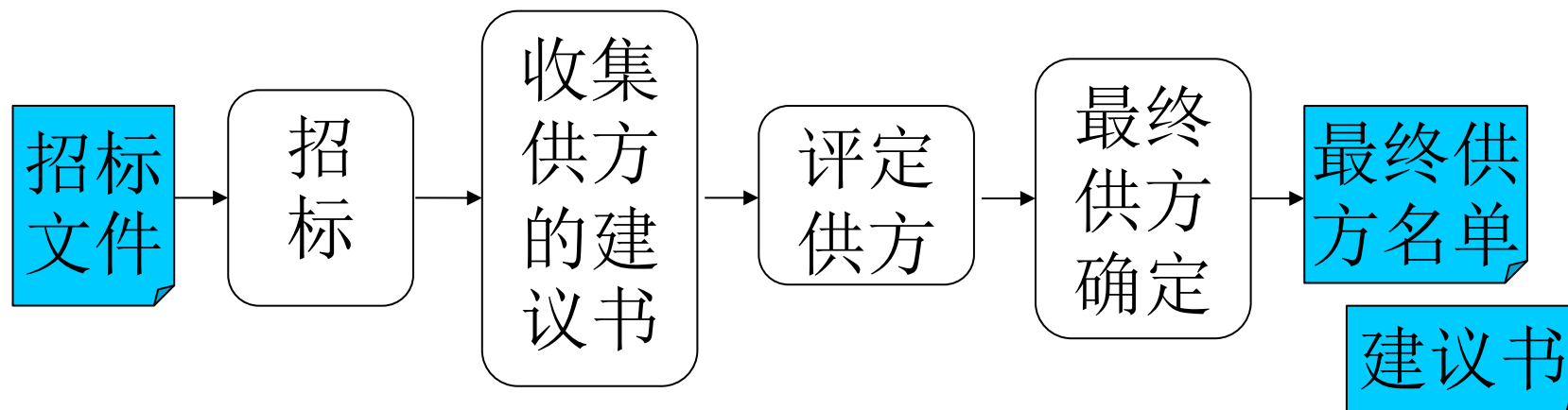


- ❑ 招标书定义(采购需求定义)
- ❑ 供方选择
- ❑ 合同签署

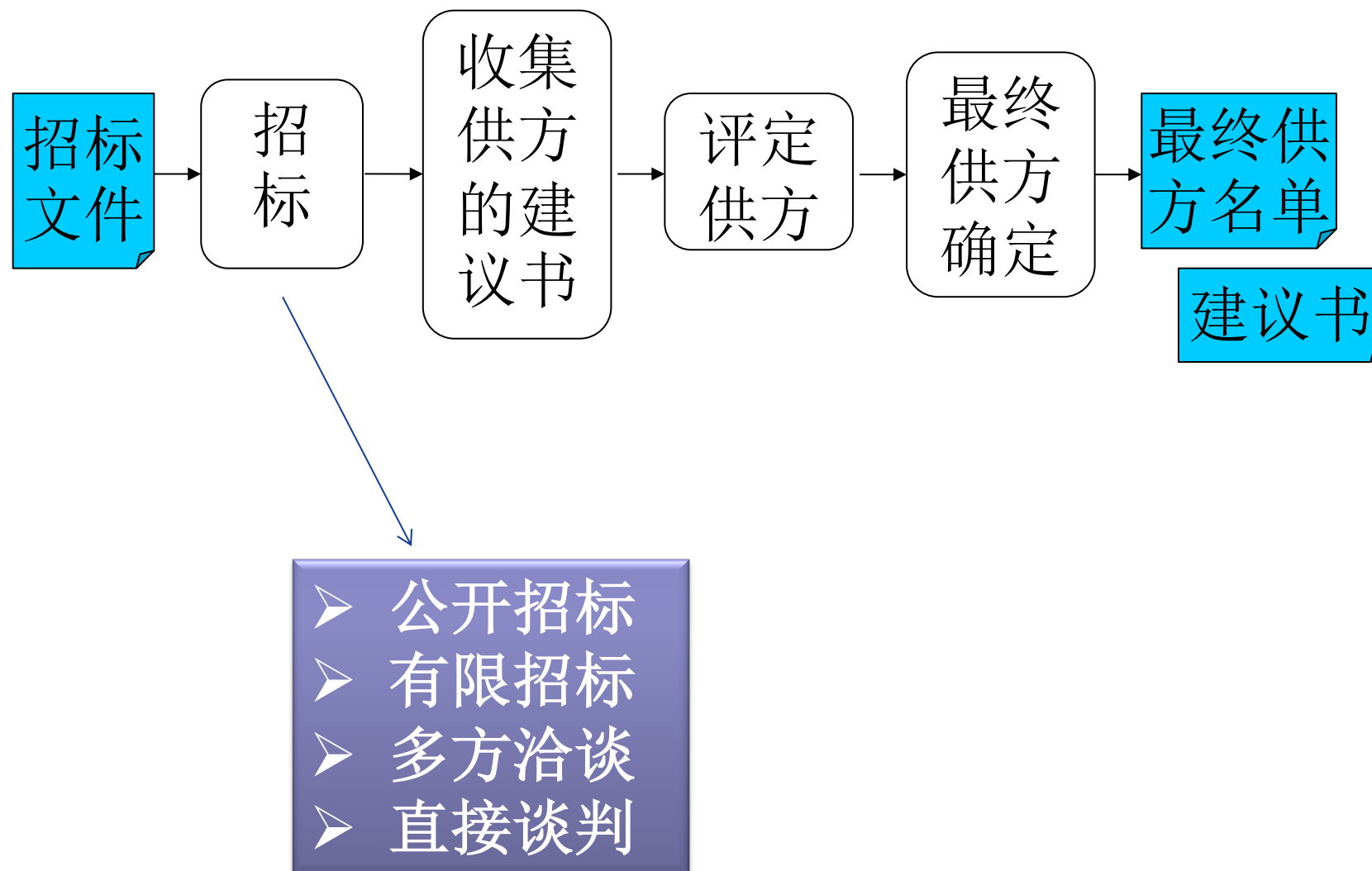
# 1、招标书定义



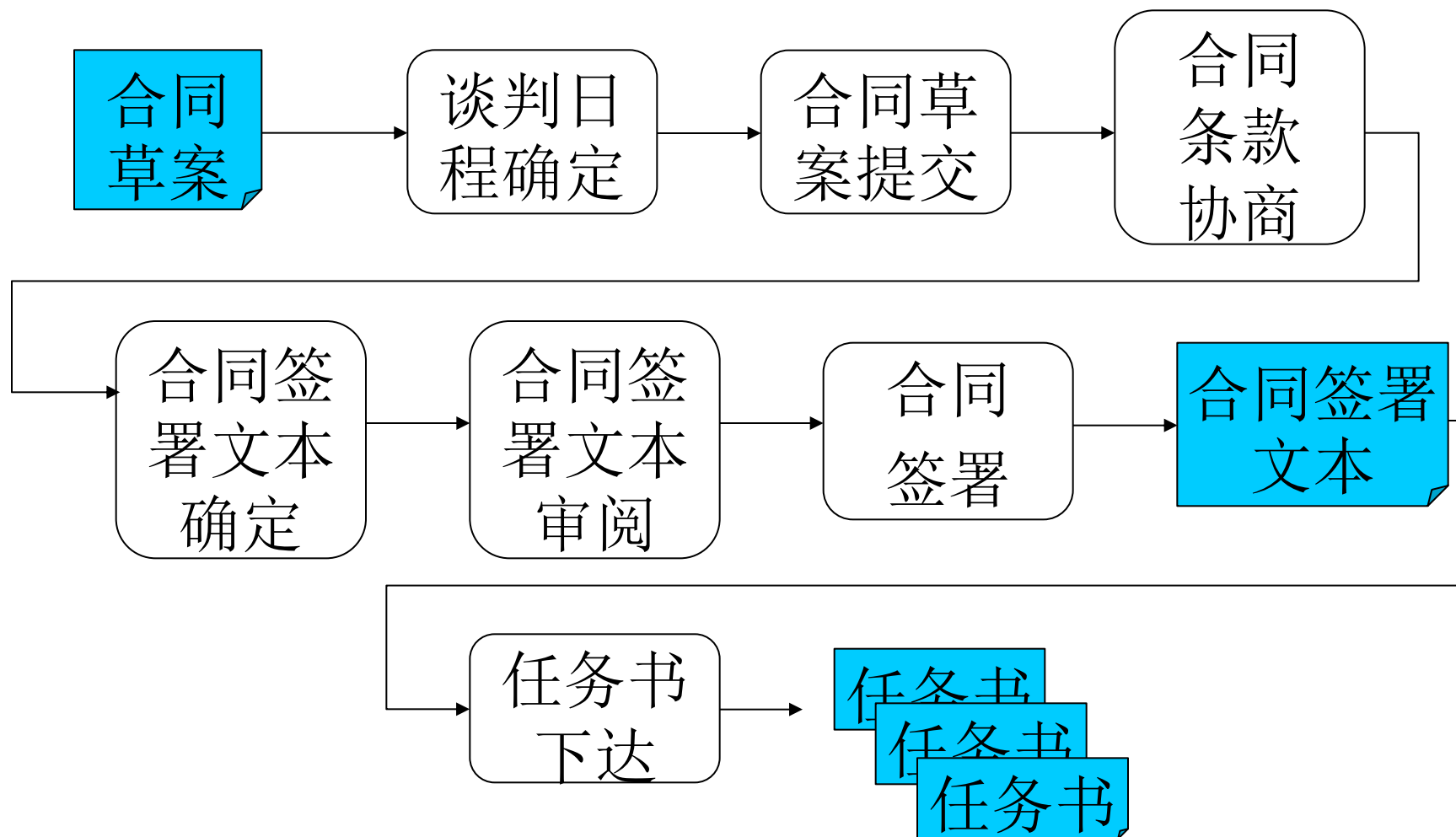
## 2、供方选择



## 2、供方选择



### 3、合同签署

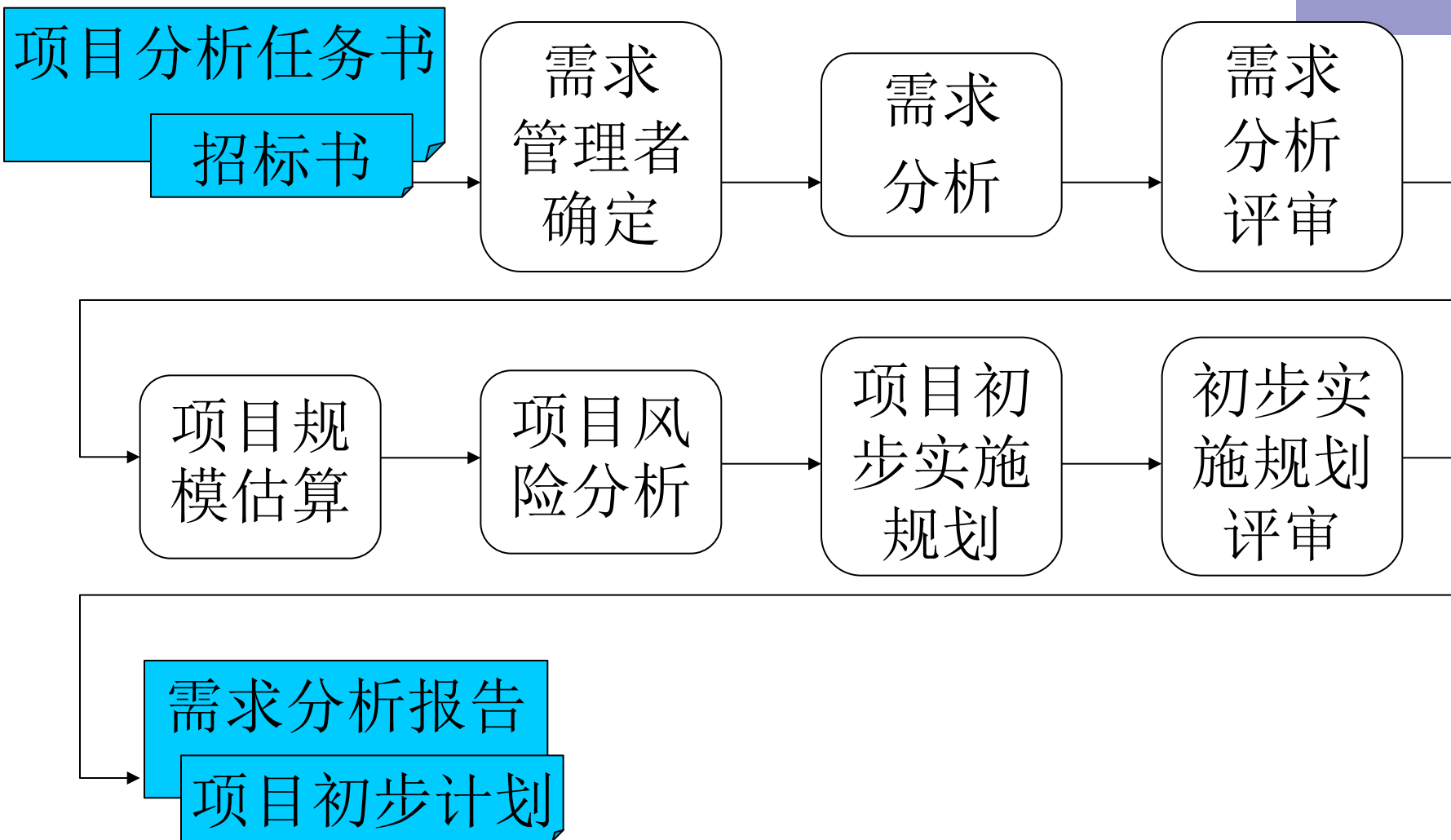


# 乙方合同环境

MIMA

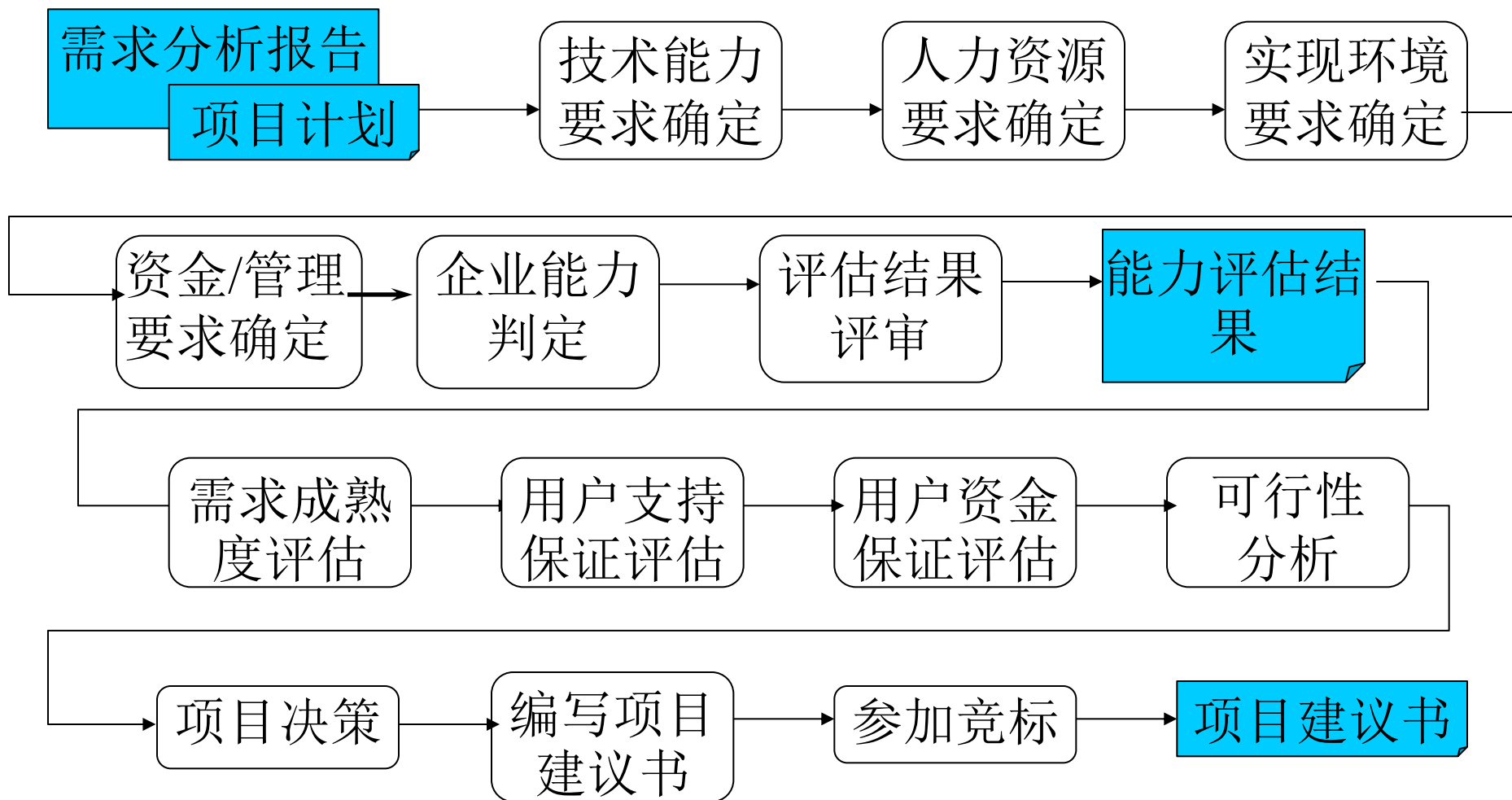
- ❑ 项目分析
- ❑ 提交建议书
- ❑ 合同签署

# 1、项目分析

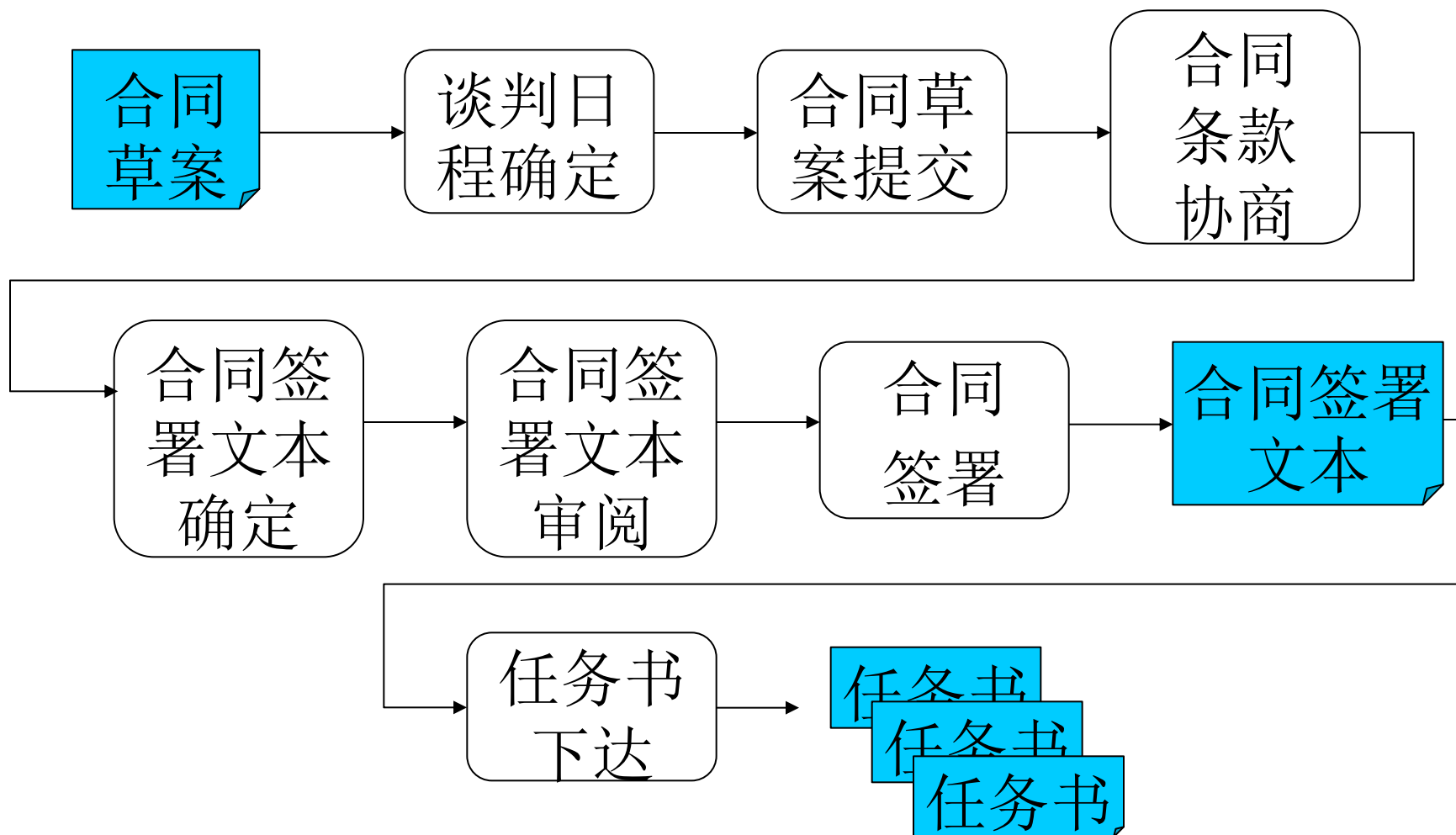




## 2、提交建议书



### 3、合同签署



- ❑ 企业内部项目实施的核心是确定任务范围和相关各方进行有效地配合。这将通过相关各方之间的协议来调整。因此，在内部项目实施中，仅仅在合同签署过程中定义了一个协议签署过程。此处协议可视为作为“合同”，但无特别的商业约束。其它方面可参考甲乙方的过程。

- ❑ 一、项目立项
- ❑ 二、授权项目
- ❑ 三、初始项目范围分析
- ❑ 四、生存期模型
- ❑ 五、案例分析

- 对项目进行授权和初始化，以便确认相关的人知晓这个项目
- 形式：文档化输出，主要是项目章程

# 项目章程(Project Charter)

MIMA

确认项目存在的文件，包括对项目的确认、对项目经理的授权和项目目标的概述等。

# Sample Project Charter

表 2-4 校务通项目的项目章程

项目名称		校务通管理系统	项目标识	QTD - SCHOOL
下达人		项目委员会	下达时间	2003 年 4 月 10 日
项目经理		韩万江	项目计划提交时限	2003 年 4 月 14 日
送达人		× × ×		
项目目标		1. 为 × × × 提供基于 B/S 结构的校务管理系统； 2. 为 × × × 提供多平台的交流		
项目范围	项目性质	公司外部项目，属于软件开发类		
	项目组成	见项目输入		
	项目要求	见项目输入		
	项目范围特殊说明	无		
项目输入		1. 《校务通管理系统实施方案建议书》； 2. 合同及其附件		
项目用户		× × × 教育委员会		
与其他项目关系		无		
项目限制	完成时间	预计完成时间为 2003 年 6 月 20 日		
	资金	见项目输入 1 第 6 章		
	资源	依据批准的项目计划		
	实现限制	B/S 结构，开发平台为 Windows NT、IIS Server、SQL Server、J2EE		

# Sample Project Charter (continued)

表 2-3 IT 升级项目的项目章程

项目题目：IT 升级项目

项目开始时间： 2008. 3. 10 项目结束始时间： 2008. 5. 15

项目经理： 赵强, Zhaoqiang@ abc. com

项目目标：根据新的企业标准对企业所有人员的软件硬件进行升级，新的标准见附件。升级可能影响服务器、一些计算机及一些网络的软硬件。软硬件的费用为 300 万元，人工成本为 20 万元。

建议方法：

- 升级数据库目录。
- 做详细的成本估算，然后上报。
- 获取软硬件报价。
- 尽可能由内部的人员参与项目。

人员	角色	职责
赵强	项目经理	规划、监控项目
王立	质量经理	负责项目的质量
江明	技术经理	负责技术
章溢	系统支持	负责产品的所有系统、网络
韩斌	采购经理	负责采购软硬件

签字：

注释：



1. 项目组织的领导者
2. 项目组织的管理者
3. 项目组织的决策者
4. 项目组织的分析者
5. 项目组织的计划者
6. 项目组织的控制者
7. 项目组织的组织者
8. 项目组织的评价者
9. 项目组织的协调者

1. 开发计划
2. 组织实施
3. 项目控制

- ❑ 一、项目立项
- ❑ 二、授权项目
- ❑ 三、初始项目范围分析
- ❑ 四、生存期模型
- ❑ 五、案例分析

## ■ 项目范围的主要内容

- 1. 项目的合理性说明
- 2. 项目目标
- 3. 项目可交付成果

## ■ 项目范围的依据

- 合同
- 规范
- SOW: 客户份额（工作说明书）

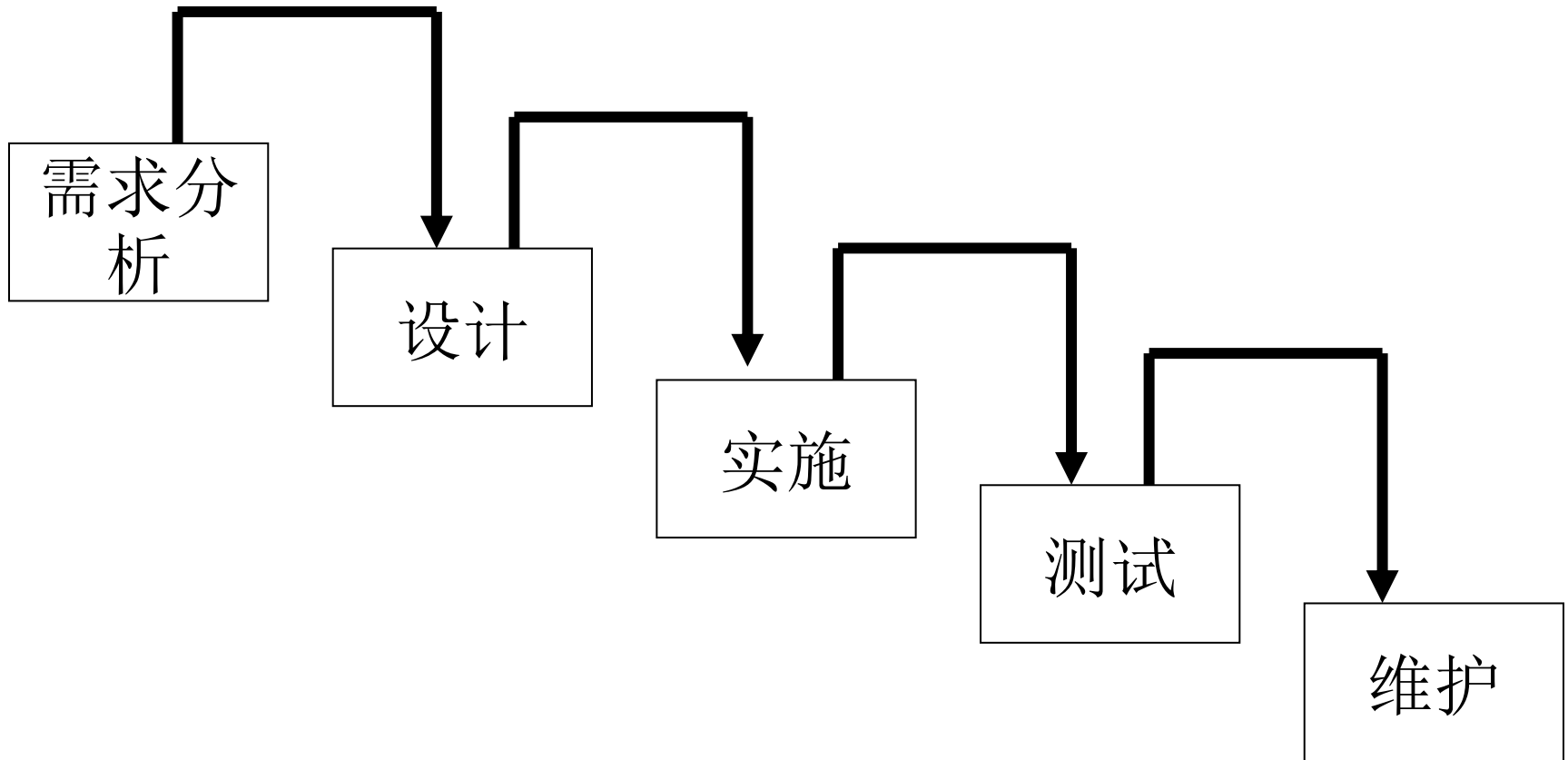
# 明确项目范围的重要性

- 后期维护是否属于软件开发项目范围必须在项目责任书里明确。
- 不属于项目范围以内的事，尽量不要做，搞不好是费力不讨好。

- ❑ 一、项目立项
- ❑ 二、授权项目
- ❑ 三、初始项目范围分析
- ❑ 四、生存期模型
- ❑ 五、案例分析

- ❑ 瀑布Waterfall
- ❑ V模型V-shaped
- ❑ 原型Prototyping
- ❑ 增量Incremental
- ❑ 螺旋式Spiral
- ❑ 快速应用开发RAD
- ❑ 渐近式阶段

# WaterFall model

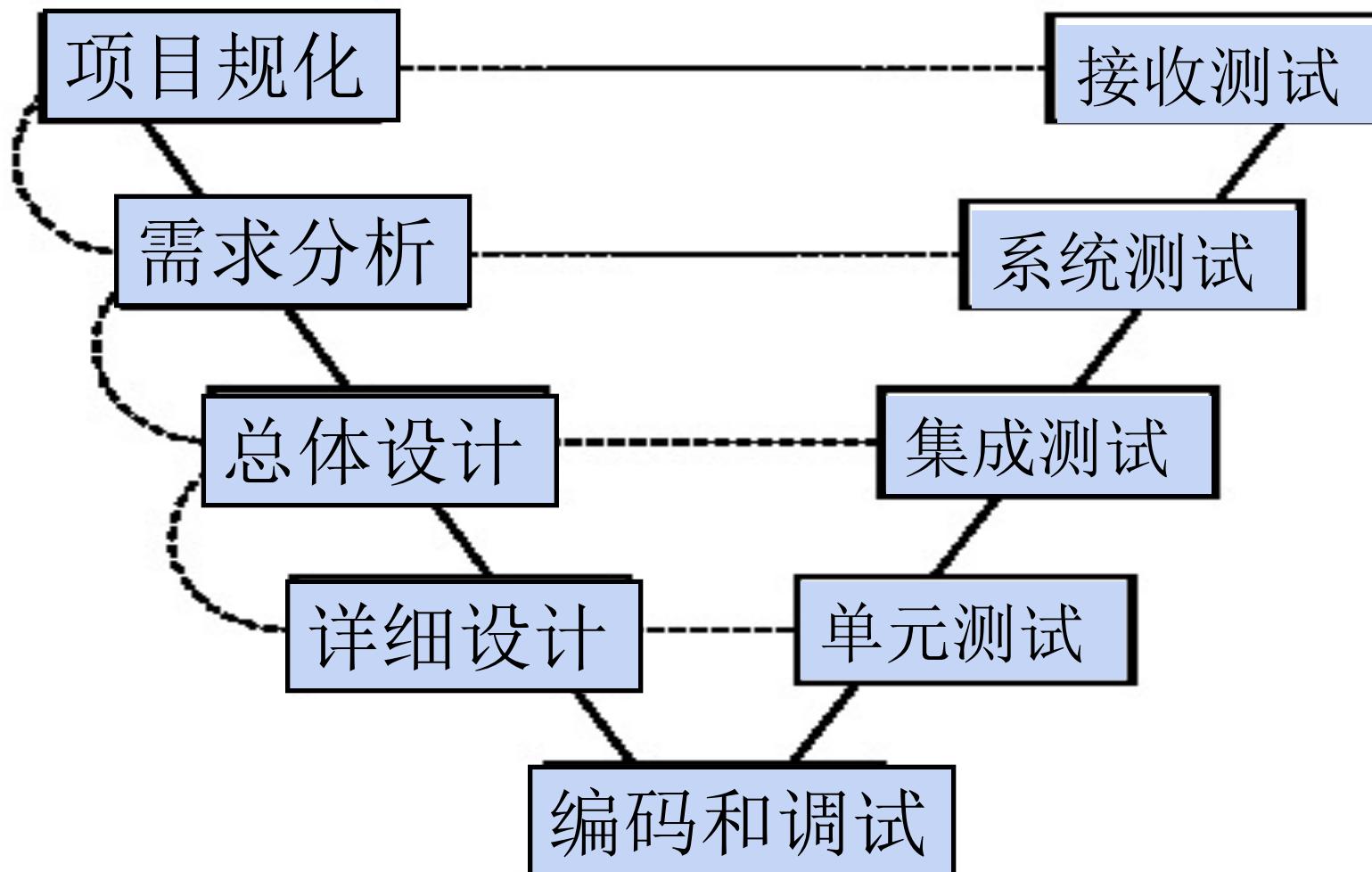


or conventional model



# WaterFall model 适合的项目

- ❑ 在项目开始前, 项目的需求很明确
- ❑ 在项目开始前, 解决方案也很明确
- ❑ 类似的项目如:
  - ❑ 公司的财务系统
  - ❑ 库存管理系统
  - 短期项目



# V模型适合的项目

- ❑ 在项目开始前, 项目的需求很明确
- ❑ 在项目开始前, 解决方案也很明确
- ❑ 对系统的性能安全很严格的项目
- ❑ 类似的项目如:
  - ❑ 航天飞机等
  - ❑ 公司的财务系统

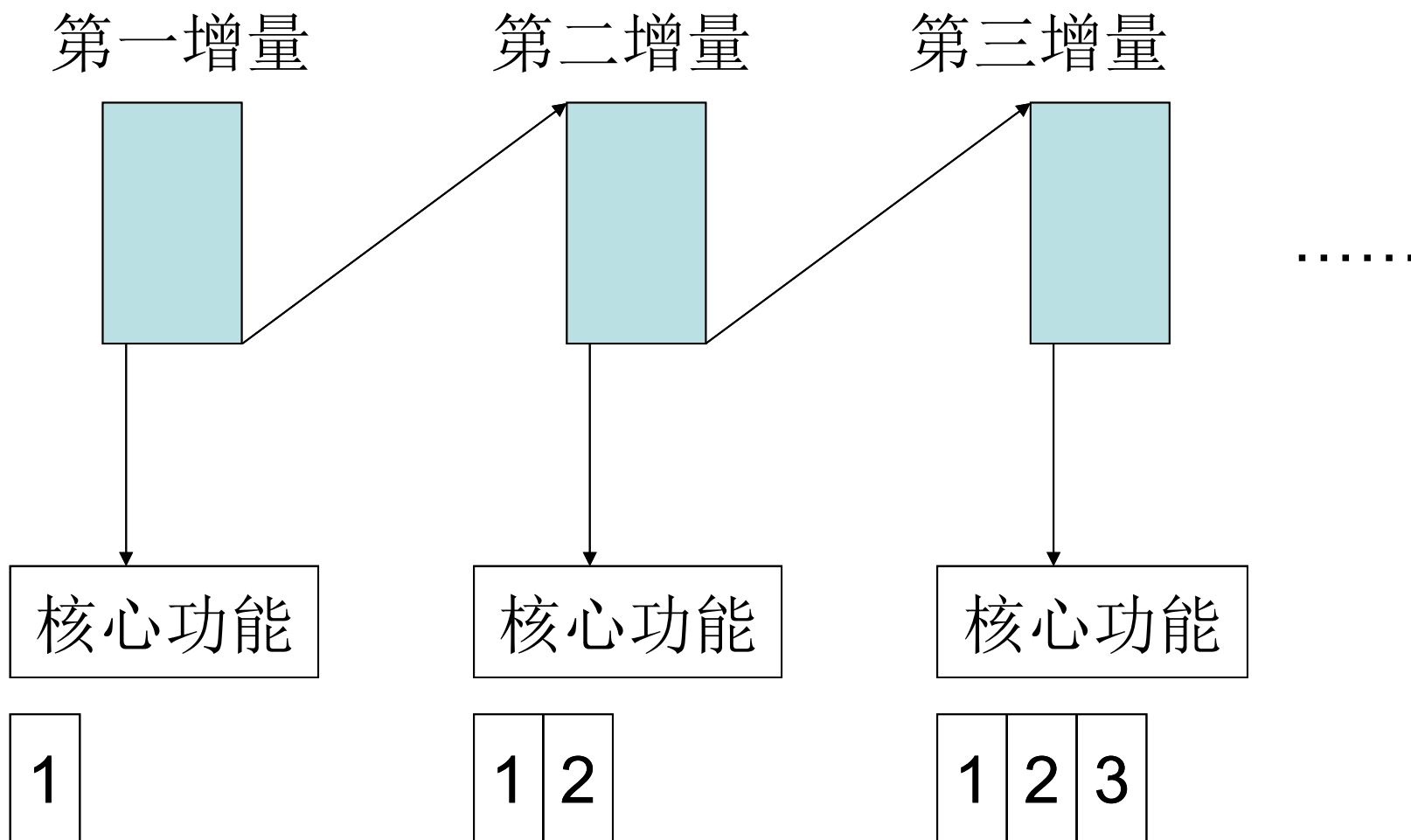
- 从最核心的方面开始，向用户展示完成的部分，然后根据用户的反馈信息继续开发原型，并重复这一过程
- 直到开发者和用户都认为原型已经足够好，然后在此基础上开发客户满意的软件产品

# Prototype 模型适合的项目

- ❑ 在项目开始前, 项目的需求不明确
- ❑ 需要减少项目需求的不确定性
- ❑ 类似的项目如:
  - ❑ 确定显示界面
  - ❑ 第一次开发的产品, 验证可行性

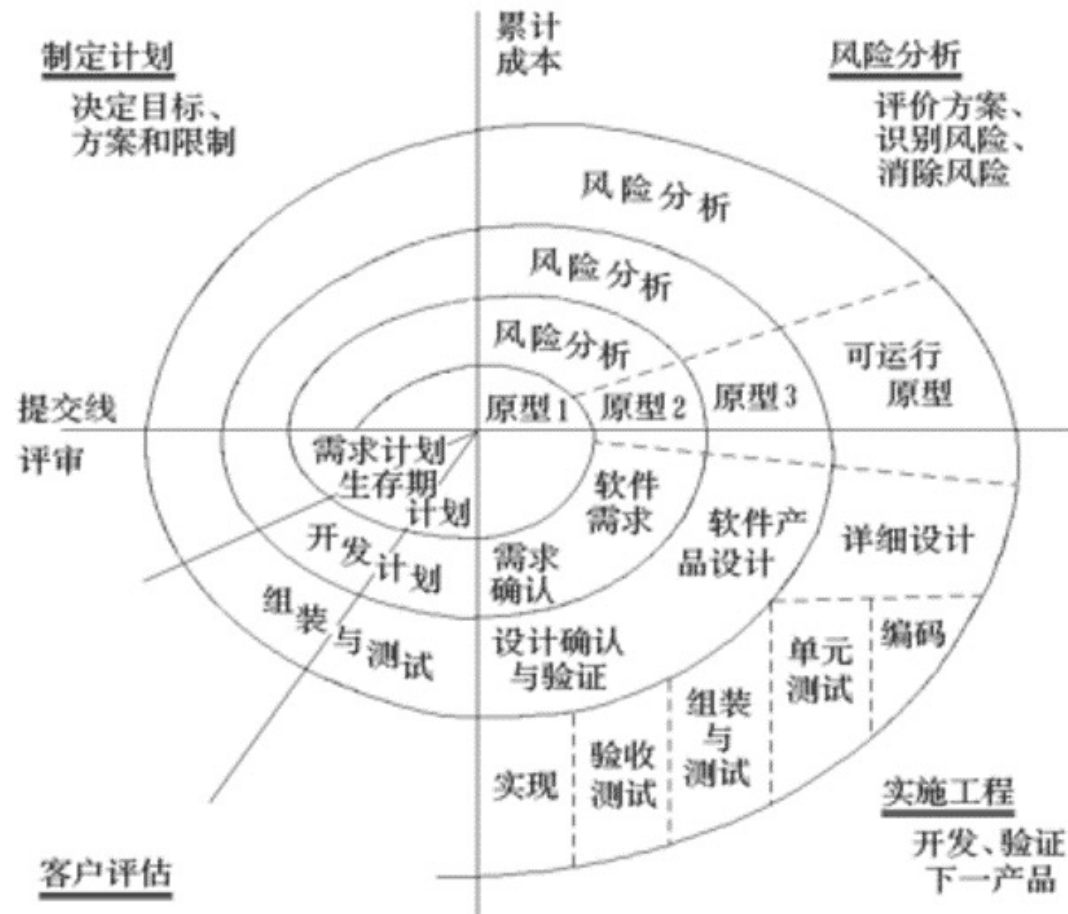
# Incremental Model 增量式模型

MIMA



- ❑ 项目开始，明确了需求的大部分，但是需求可能会发生变化
- ❑ 对于市场和用户把握不是很准，需要逐步了解
- ❑ 对于有庞大和复杂功能的系统进行功能改进，就需要一步一步实施的。

# Spiral Model 螺旋式模型





- ❑ 螺旋模型沿着螺线旋转，在四个象限上分别表达了四个方面的活动，即：
- ❑ **制定计划**——确定软件目标，需求和选定实施方案，弄清项目开发的限制条件
- ❑ **风险分析**——评估所选方案，考虑如何识别和消除风险
- ❑ **实施工程**——实施软件开发，编码，测试等
- ❑ **客户评估**——评价开发工作，提出修正建议，规划下期任务

# Spiral Model 适合的项目

- ❑ 风险是主要的制约因素
- ❑ 不确定因素和风险限制了项目进度
- ❑ 用户对自己的需求也不是很明确
- ❑ 需要对一些基本的概念进行验证
- ❑ 可能发生一些重大的变更
- ❑ 项目规模很大
- ❑ 项目中采用了新技术

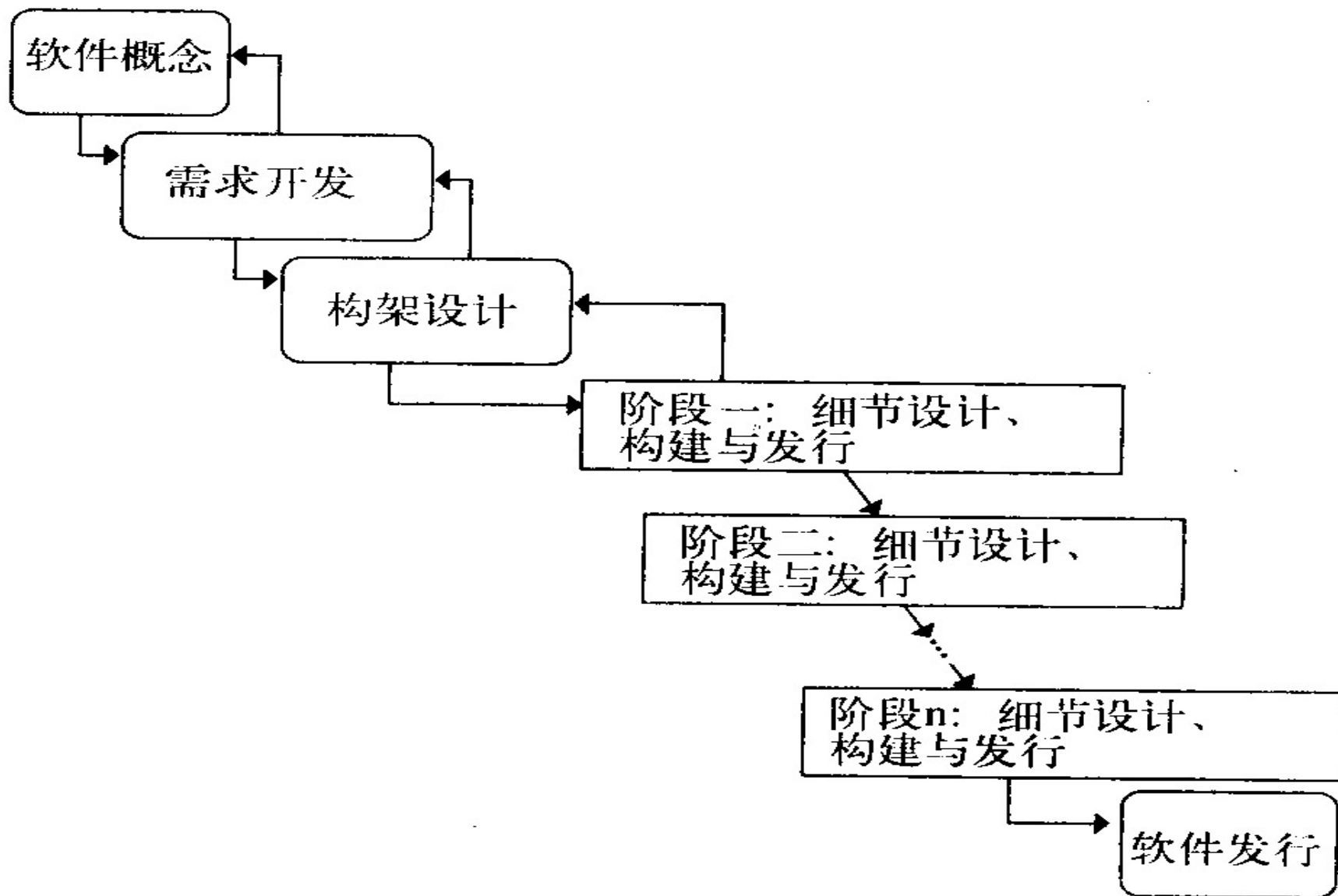
# 最常用的一渐进式阶段模型

综合了增量模型和螺旋式模型的一个实用模型

- 渐进式前进
- 阶段式提交

# 阶段性完成规划

MIMA



# 渐进式阶段模型的特点

- ❑ 阶段式提交一个可运行的产品, 每个阶段提交的产品是独立系统
- ❑ 关键的功能更早出现, 可提高开发人员和客户信心
- ❑ 早期预警问题, 避免软件缺陷不知不觉的增长
- ❑ 阶段式提交产品说明进展, 减少报告负担
- ❑ 阶段性完成可以降低估计失误
- ❑ 阶段性完成均衡了弹性与效率

# 渐进式阶段模型适合的项目

- ❑ 可以适合任何规模的项目，主要是中型或大型项目
- ❑ 希望随时看到未来的项目

# 选择生存期的步骤

- ❑ 熟悉各种生存期模型
- ❑ 评审、分析项目的特性
- ❑ 选择适合项目的生存期模型
- ❑ 标识生存期模型与项目不一致地方, 并进行裁减

- ❑ 一、项目立项
- ❑ 二、授权项目
- ❑ 三、初始项目范围分析
- ❑ 四、生存期模型



- ❑ 项目初始过程
  - ❑ 一、项目来源
  - ❑ 二、授权项目
  - ❑ 三、初始项目分析
  - ❑ 四、生存期模型

# 生存期模型总结

- 1.在前期需求明确的情况下尽量采用瀑布模型或改进型的瀑布模型.
- 2.在用户无信息系统使用经验,需求分析人员技能不足情况下一定要借助原型.
- 3.在不确定性因素很多,很多东西前面无法计划情况下尽量采用增量迭代和螺旋模型
- 4.在需求不稳定情况下尽量采用增量迭代模型
- 5.在资金和成本无法一次到位情况下可以采用增量模型,软件产品分多个版本进行发布
- 6.对于完全多个独立功能开发可以在需求阶段就分功能并行,但每个功能内都应该遵循瀑布模型
- 7.对于全新系统的开发必须在总体设计完成后再开始增量或并行.
- 8.对于编码人员经验较少情况下建议不要采用敏捷或迭代等生命周期模型.
- 9.增量,迭代和原型可以综合使用,但每一次增量或迭代都必须有明确的交付准则