

# 软件工程

Software Engineering

## 第2章 可行性研究

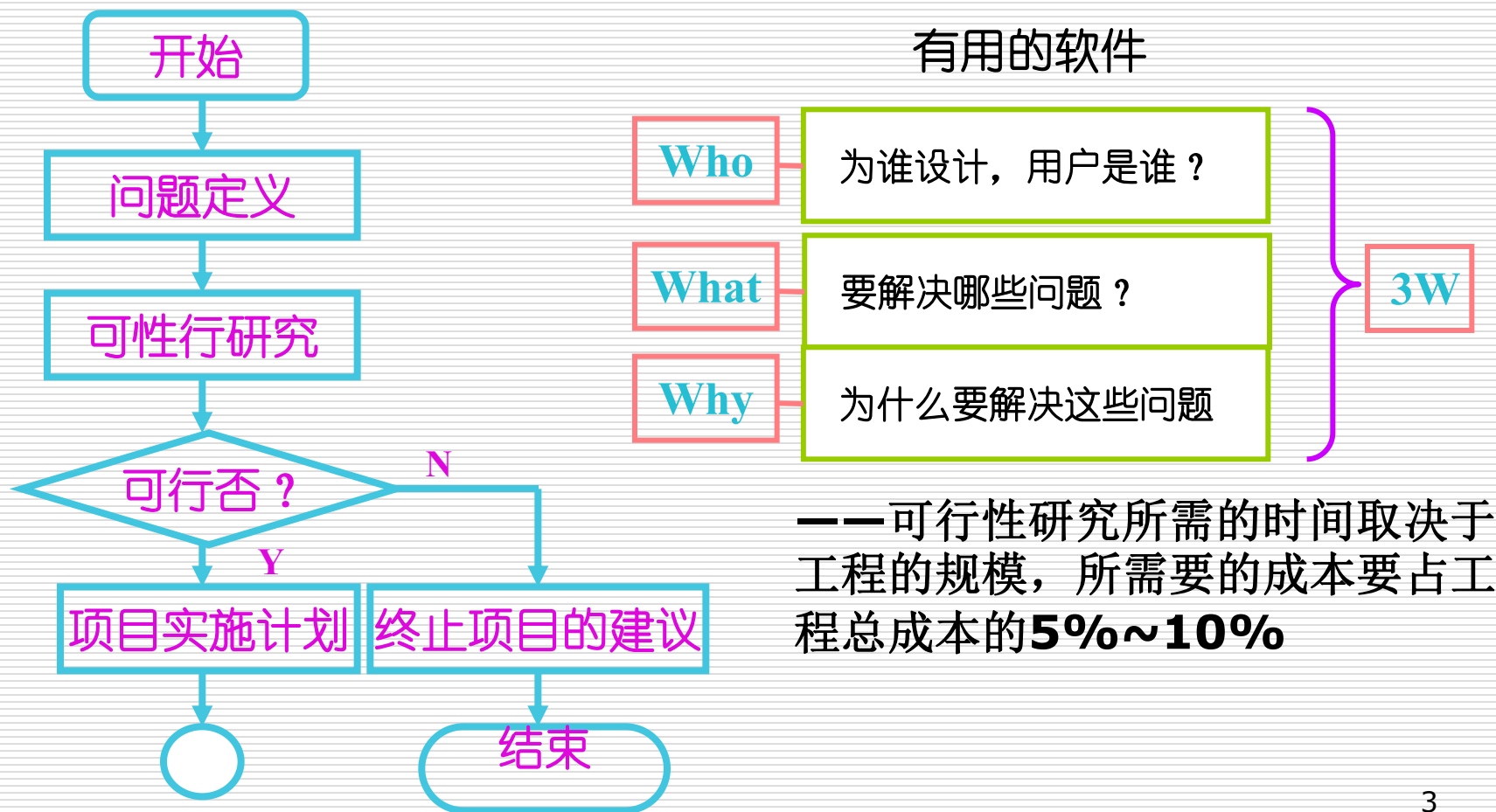
欧毓毅

- 在客观世界中，并不是所有的问题都可以有明显的解决方法；
- 如果这些问题没有行得通的解决办法，那么贸然开发这些项目就会造成时间、人力、资源和经费的巨大浪费。
- 同样，对软件的项目开发也存在这一问题，所以，必须对开发项目进行可行性研究。

说明该软件开发项目的实现在技术上、经济上和社会条件上的可行性；评述为合理地达到开发目标可能选择的各种方案。  
。（GB 8567-88 《计算机软件产品开发文件编制指南》）

# 问题的定义与可行性研究

- 可行性研究的目的是不是解决问题，而是确定问题是否**值得**去解决；
- 可行性研究的目的是用**最小的代价**在尽可能**短的时间**内确定问题是否能够**解决**；
- 可行性研究实质上是要进行一次大大压缩**简化**了的系统分析和设计的过程，也就是在较高层次上以比较抽象的方式进行的系统分析和设计的过程。



# 主要内容

## 2.1 可行性研究的任务

## 2.2 可行性研究过程

## 2.3 系统流程图

## 2.4 数据流图

## 2.5 数据字典

## 2.6 成本/效益分析

## 2.1 可行性研究的任务

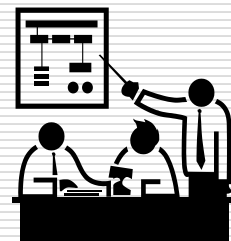
可行性研究的主要任务是“了解客户的要求及现实环境，从技术、经济和社会因素等三方面研究并论证本软件项目的可行性，编写可行性研究报告，制定初步项目开发计划。”

GB 8566-88 《计算机软件开发规范》

可行性研究的目的是不是解决问题，而是确定问题是否值得去解决。

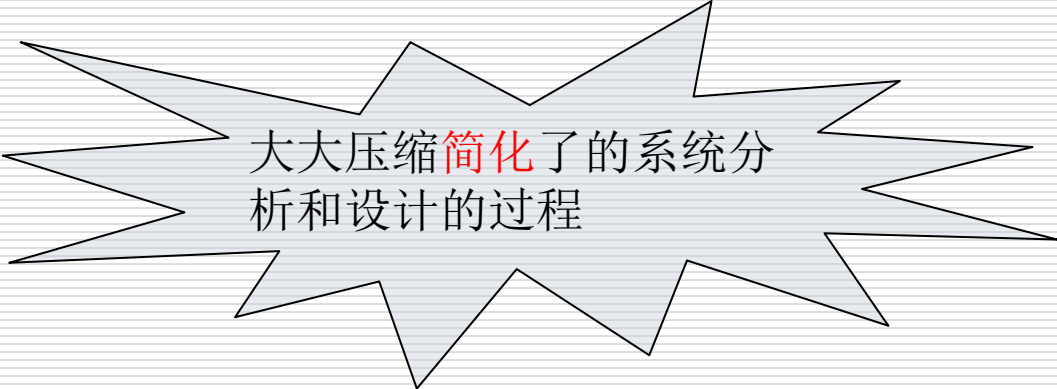
可行性研究分析过程：

- 进一步分析和澄清问题定义
- 分析员应该导出系统的逻辑模型
- 探索若干种可供选择的主要解法



# 可行性研究的内容

- **1.**进一步分析和澄清问题的定义，需要进行概要的分析研究，初步确定项目的**规模和目标**，确定项目的**约束和限制**，把它们清楚地列举出来。
- **2.**导出系统的逻辑模型；
- **3.**探索若干种可供选择的解法（系统实现方案）；



大大压缩**简化**了的系统分析和设计的过程

# 可行性研究的内容

至少应该从下述4个方面研究每种解法的可行性

**技术可行性**使用现有的技术能实现这个系统吗？

**经济可行性**这个系统的经济效益能超过它的开发成本吗？

**法律可行性**法律方面和使用方面能符合现有的律法和规章吗？

**操作可行性**系统的操作方式在这个用户组织内行得通吗？

# 可行性分析

## □ 一、技术可行性研究。

■ 度量一个特定技术信息系统解决方案的实用性及技术资源的可用性，即：根据客户提出的系统功能、性能及实现系统的各项约束条件，从技术的角度研究实现系统的可行性。例如：

- 使用现有的技术能否实现这个系统？
- 是否有胜任开发该项目的熟练技术人员？
- 能否按期得到开发该项目所需的软件、硬件资源？



- 技术可行性一般要考虑的情况如下：
- （1）技术。
  - 相关技术的发展，现有技术能否实现新系统，技术难点、建议采用技术的先进性，通过调查了解当前最先进的技术，分析相关技术的发展是否支持这个系统。
- （2）资源的有效性。

用于建立系统的硬件设备、软件、开发环境等资源是否具备。特别是用于开发项目的人员在技术和时间上是否存在问题。
- （3）开发风险分析

# 技术可行性可以使用的工具

- **数学建模、原型建造和模拟**是基于计算机系统技术分析活动的有效工具，描述了技术分析建模过程的信息流图。
- 系统分析员通过对现实世界的观察和分析建立技术分析**模型**，评估模型的行为并将它们与现实世界对比，论证**系统**开发在技术上的可行性和优越性。
- 基于计算机系统**模型**必须具备下列特性：
  - （1）能够反映系统配置的**动态特性**，容易理解 and 操作，能够提供系统**真实的结果**并有利于评审。
  - （2）能够综合与系统有关的**全部因素**，能够再现系统运行的结果。
  - （3）能够突出与系统有关的**重要因素**，能够忽略与系统无关的或次要的因素。
  - （4）结构简单，容易实现，容易修改。

## □ 二、经济可行性研究。

■ **度量系统解决方案的性能价格比**，即进行成本效益分析，评估项目的开发成本，估算开发成本是否会超过项目预期的全部利润。分析系统开发对其他产品或利润的影响。

■ **所要考虑的问题是：这个系统的经济效益能否超过它的开发成本？要开发的项目是否值得投资开发？**

成本/效益分析（开发、运行的成本/效益）

■ 有形成本、效益

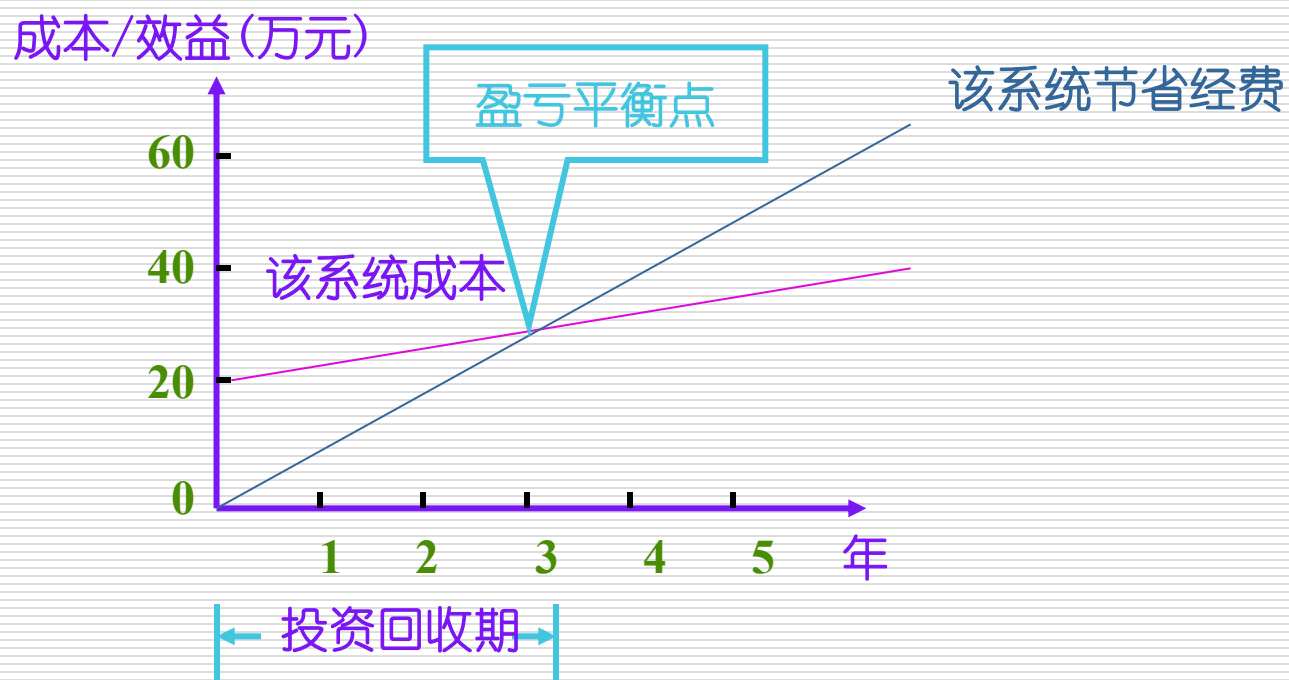
■ 无形成本、效益

价值和成本的关系

■ 质量与价值、成本的关系

■ 价值/成本的均衡

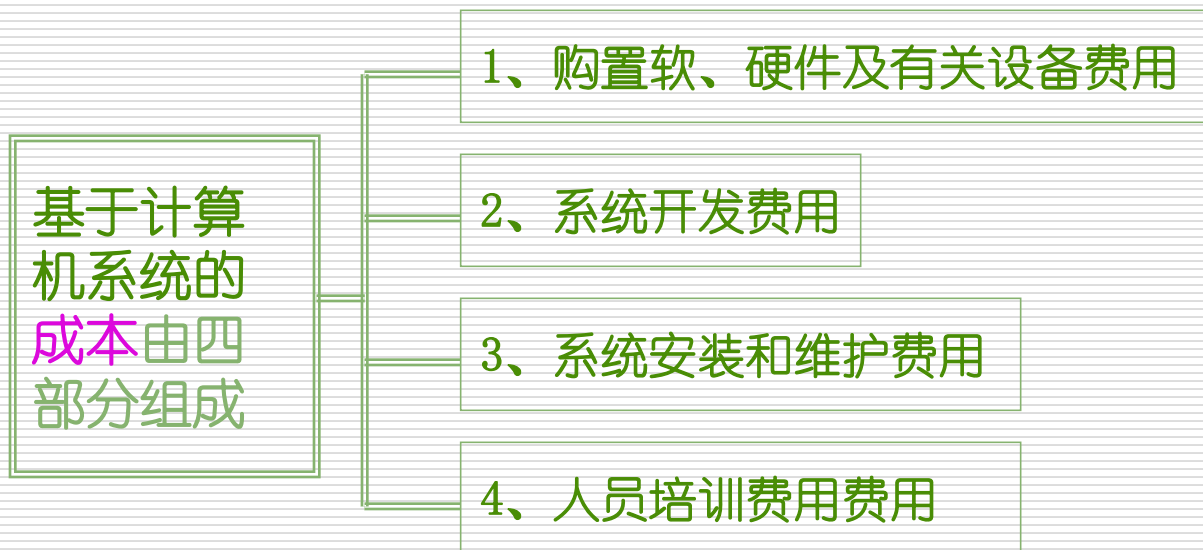
# 举例



----成本及效益分析图

- 经济可行性问题一般包含两方面：一方面是**经济实力**；另一方面是**经济效益**。

一般说来，基于**计算机系统的成本**由如下四部分组成



但现实中的成本还包括许多其他方面；

- **预估一下**：做一个正规的中小型**app**要多少钱（半年）？

## 工资：

- 一个成熟的团队，团队的结构以架构师兼后台开发**1名+安卓1名+iOS开发1名+产品经理1名+前端开发1名+设计师1名+2个浮动名额**
- 假设团队平均工资在**13000**左右

缴存项目	个人部分		单位部分	
养老	8%	1040.00	14%	1820.00
医疗	2%	260.00	6.2%	806.00
失业	1%	130.00	2%	260.00
工伤			0.4%	52.00
生育			0.5%	65.00
公积金	<input type="text" value="5"/> %	650.00	5%	650.00
合计	16%	2080.00	36.1%	3653.00

除工资的部分，还要平均为每个员工多缴纳 **3653**元的五险一金支出，员工部分的支出是  $(13000+3653)*8=133224$ 元/月。

100平米中低端办公场所+水电、物业：一个月15000左右。

办公耗材、宽带、桌椅等等：6000

产品定位、扯皮、早期工作准备阶段：产品定位、早期的需求、需求调研、竞品调研分析、注册域名、购置服务器以及周边基础能力（如CDN存储、短信服务、消息推送等）、商标（未注册商标）、抢注域名、设计logo、原型、技术架构：10000元/年。

产品版本策划、架构设计阶段：产品官网、用户协议、种子用户、基础数据上报规划、UI设计与交互标准制定、后台管理系统、APP.....

内部测试、打磨阶段：收集问题、发现问题、优化、解决问题，期间可能需要发布1-3个beta子版本。想办法保持种子用户的活跃与配合的激情。

发布阶段：不算渠道宣传费用，不打广告，上线iOS APP Store 需要注册开发者账号，购买证书：99美元/年。找安全团队做安全评估，公关费.....

6个月的人工费用：133224元\*6个月  $\approx 80$  w。场地租金：15000元\*6个月 = 9w。办公耗材+桌椅支出：6000元\*6个月。服务器、域名、周边IT基础费用：2w。

- 在系统分析和设计阶段只能得到上述费用的预算，即估算成本。在系统一切完毕并交付用户运行后，上述费用的统计结果就是实际成本。
- 系统效益包括**经济效益**和**社会效益**两部分。
  - 经济效益指应用系统为用户增加的收入，它可以通过直接的或统计的方法估算；
  - 社会效益只能用定性的方法估算。

### □三、社会可行性(法律可行性)

- 主要讨论法律方面和使用方面的可行性。研究在系统开发过程中可能涉及的各种合同、侵权、责任以及各种与法律相抵触的问题。例如，被开发软件的权利归属问题、软件所使用的技术是否会造成侵权等问题。

### □四、操作可行性\*\*\*

- 操作可行性评价系统运行后会引起的各方面变化，
  - 用户使用可能性
  - 时间进度可行性
  - 组织和文化上的可行性

□如：对组织机构管理模式、用户工作环境等产生的影响。



# 第2章可行性研究

2.1 可行性研究的任务

2.2 可行性研究过程

2.3 系统流程图

2.4 数据流图

2.5 数据字典

2.6 成本/效益分析

## 2.2 可行性研究过程

怎样进行可行性研究呢?典型的可行性研究过程有下述8个步骤。

- 1. 复查系统**规模**和**目标**
- 2. 研究目前正在使用的**系统**
- 3. 导出新系统的**高层逻辑模型**
- 4. 进一步定义**问题**
- 5. 导出和评价供选择的**解法**
- 6. 推荐**行动方针**
- 7. 草拟开发**计划书**
- 8. 写文档提交**审查**

## 2.2 可行性研究过程

- 1. 系统定义，复查系统规模和目标
  - **系统定义**是一个系统的关键，如果系统没有定义好，也就是没有确定系统的边界。就谈不上确定项目规模和目标。
  - 为了定义好一个系统，分析员对有关人员进行调查访问，仔细阅读和分析有关材料，复查系统规模和目标，对项目的规模和目标进行定义和确认，改正含糊或不确切的叙述，**清晰**描述项目的一切限制和约束，确保分析员正在解决的问题确实是要解决的问题。

### 系统目标与规范说明书

- 1、项目：教材销售系统
- 2、问题：人工销售教材手续繁琐，容易出错
- 3、项目目标：建立一个高效率、无差错的计算机教材销售系统
- 4、项目范围：利用现有的计算机，软件开发费用不得超过2000元
- 5、初步想法：建议在系统中增加对缺书的统计与采购功能
- 6、可行性研究：建议进行大约10天的可行性研究，且研究费用不超过500元

## □ 2. 对于现行系统进行分析研究

- 要研究目前正在使用的系统。如果目前有一个系统正在使用，那么这个系统一定能完成某些有用的工作。所以新系统的目标也必须能完成这些基本功能。如果现有的系统是完美的，那么用户就不会提出开发新系统。
- 要认识到现行系统是信息的重要来源。需要研究它的基本功能、性能、环境，存在的问题，运行现行系统需要多少费用，对新系统有什么新的功能要求，新系统运行时能否减少使用费用等。（不要关注实现细节）
- 具体方法可以实地考察现行系统，收集、研究和分析现行系统的文档资料。

### □ 3. 导出新系统的高层逻辑模型

- 根据对现行系统的分析研究，搞清了新旧系统的特征，逐渐明确新系统的功能、处理流程以及所受的约束。
- 有了这些理解后，就可以用建立逻辑模型的工具——数据流图和数据字典来描述数据在系统中的流动和处理情况。
- 设想目标系统的逻辑模型，最后根据目标系统的逻辑模型建造新的物理系统。

### □ 4. 进一步定义问题

- 重新定义问题，发现和改正错误；

## □ 5. 导出和评价供选择的解法（物理解决方案）

- 分析员应该从他建议的系统逻辑模型出发，导出若干个较高层次的物理解法供比较和选择。
- 其次可以考虑操作方面的可行性。分析员应该根据使用部门处理事务的原则和习惯检查技术上可行的那些方案，去掉其中从操作方式或操作过程的角度看用户不能接受的方案。
- 接下来应该考虑经济方面的可行性。分析员应该估计余下的每个可能的系统的开发成本和运行费用，并且估计相对于现有的系统而言这个系统可以节省的开支或可以增加的收入。
- 最后为每个在技术、操作和经济等方面都可行的系统制定实现进度表，这个进度表不需要制定得很详细，通常只需要估计生命周期每个阶段的工作量。

## □ 6.推荐行动方针

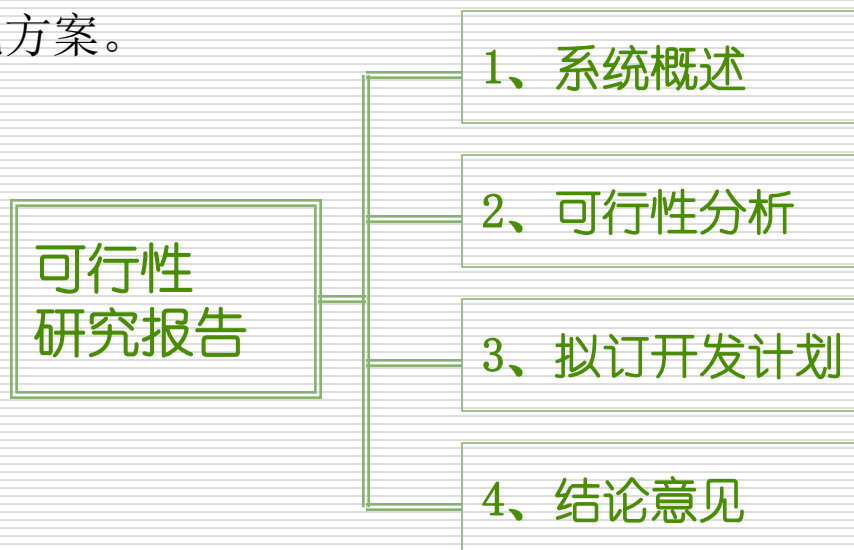
- 根据可行性研究结果应该决定的一个关键性问题是： 是否继续继续进行这项开发工程？分析员必须清楚地表明他对这个关键性决定的建议。
- 如果分析员认为值得继续进行这项开发工程，那么他应该选择一种最好的解法，并且说明选择这个解决方案的理由。
- 通常客户主要根据经济上是否划算决定是否投资于一项开发工程，因此分析员对于所推荐的系统必须进行比较仔细的成本/效益分析。

## □ 7. 草拟开发计划

- 分析员应该为所推荐的方案**草拟**一份开发计划，除了制定工程进度表之外还应该估计对各类开发人员和各种资源的需要情况，应该指明什么时候使用以及使用多长时间。此外还应该估计系统生命周期每个阶段的成本。最后应该给出下一个阶段(需求分析)的详细进度表和成本估计。

## □ 8. 编写可行性研究报告，提交审查

- 将上述可行性研究过程的结果按照以下顺序写：说明要求、目的、条件与限制、可行性研究方法及评价尺度；处理流程、工作负荷、费用开销和局限性；说明处理流程、运行环境和局限性；技术条件的可行性；经济方面的可行性；社会条件的可行性；其他可供选择的系统；结论；
- 提请用户和使用部门仔细审查，从而决定该项目是否进行开发，是否接受可行的实现方案。





# 第2章可行性研究

2.1 可行性研究的任务

2.2 可行性研究过程

2.3 系统流程图

2.4 数据流图

2.5 数据字典

2.6 成本/效益分析

可行性分析所需工具

## 2.3 系统流程图

- 系统流程图：**概括地描绘物理系统的传统工具**。
- 它的基本思想：**用图形符号以黑盒子形式**描绘组成系统的每个部件(程序，文档，数据库，人工过程等)。
- 系统流程图表达的是数据在系统各部件之间流动的情况，而不是对数据进行加工处理的控制过程，因此尽管系统流程图的某些符号和程序流程图的符号形式相同，但是它却是**物理数据流图**而不是程序流程图。

## □ 2.3.1 符号

以概括的方式抽象地描绘一个实际系统时，仅仅使用下图中列出的基本符号就足够了

符号	名称	说明
	处理	如：程序，处理机，人工加工
	输入/输出	表示输入或输出
	连接	同一页上图的连接
	换页连接	不同页上图的连接
	数据流	指明数据流动方向

图3.1 基本符号

稍微记一下!

需要更具体地描述一个物理系统时还需要使用右图中列出的系统符号

符号	名称	说明
	穿孔卡片	穿孔卡片输入/输出，或穿孔卡片文件
	文档	打印输出，或打印终端输入数据
	磁带	磁带输入/输出，或表示磁带文件
	联机存储	任何种类磁盘存储，如磁盘、磁鼓等
	磁盘	磁盘输入/输出，或磁盘上文件、数据库
	磁鼓	磁鼓输入/输出，或磁鼓上文件、数据库
	显示	显示器部件
	人工输入	人工输入数据，如填写表格
	人工操作	人工完成的处理
	辅助操作	使用辅助设备进行的脱机操作
	通信链路	通过远程通信线路传送数据

稍微记一下！

图3.2 系统符号

## □ 2.3.2 例子

### □ 一个简单的库存系统例子：

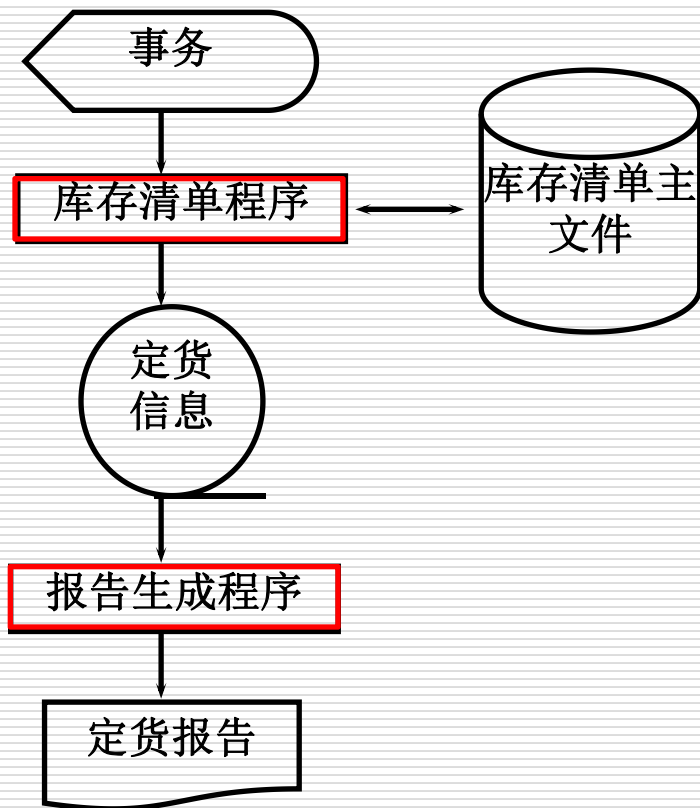
- 某装配厂有一座存放零件的仓库，仓库中现有的各种零件的数量以及每种零件的库存量临界值等数据记录在库存清单主文件中；
- 当仓库中零件数量有变化时，应该及时修改库存清单主文件，如果哪种零件的库存量少于它的库存量临界值，则应该报告给采购部门以便订货；
- 规定每天向采购部门送一次订货报告。

### 系统要求：

- 该装配厂使用一台小型计算机处理更新库存清单主文件和产生订货报告的任务。
- 零件库存量的每一次变化称为一个事务，由放在仓库中的CRT终端输入到计算机中；
- 系统中的库存清单程序对事务进行处理，更新存储在磁盘上的库存清单主文件，并且把必要的订货信息写在外存储设备上。
- 最后，每天由报告生成程序读一次外存储设备，并且打印出订货报告。

## □ 2.3.2 例子

库存系统流程图（一种）：



另一种常见的系统流程图：

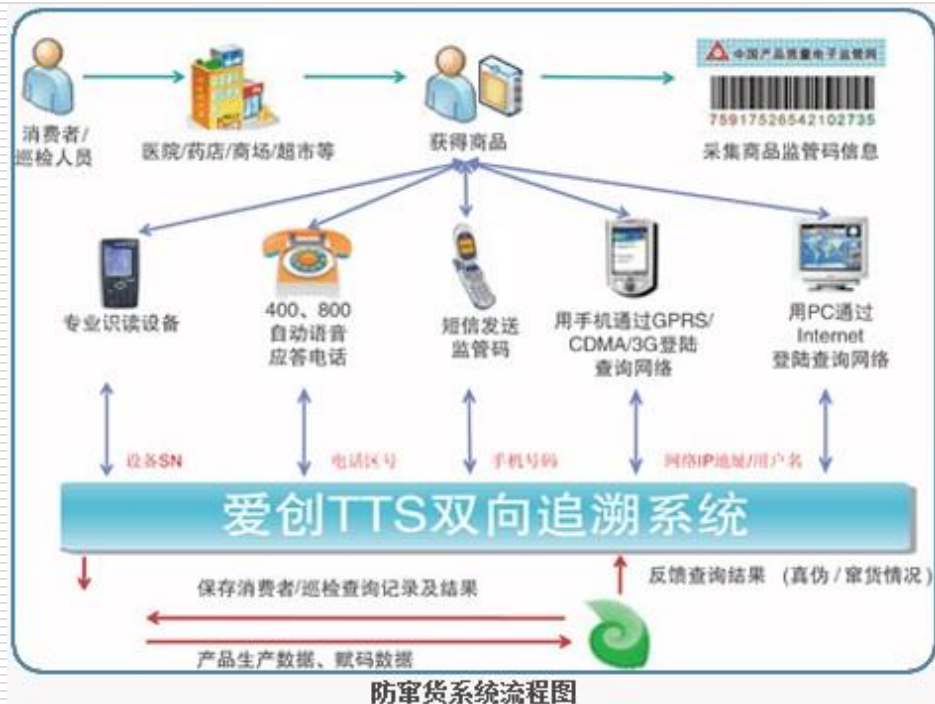


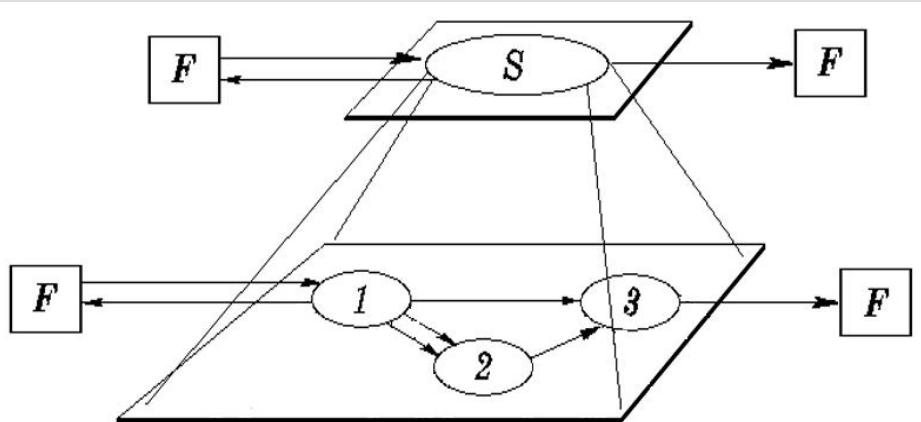
图 库存清单系统的系统流程图

### □ 2.3.3 分层

面对复杂的系统时，一个比较好的方法是分层次地描绘这个系统。

首先用一张高层次的系统流程图描绘系统总体概貌，表明系统的关键功能。然后分别把每个关键功能扩展到适当的详细程度，画在单独的一页纸上。

这种分层次的描绘方法便于阅读者按从抽象到具体的过程逐步深入地了解一个复杂的系统。



# 主要内容

2.1 可行性研究的任务

2.2 可行性研究过程

2.3 系统流程图

2.4 数据流图

2.5 数据字典

2.6 成本/效益分析

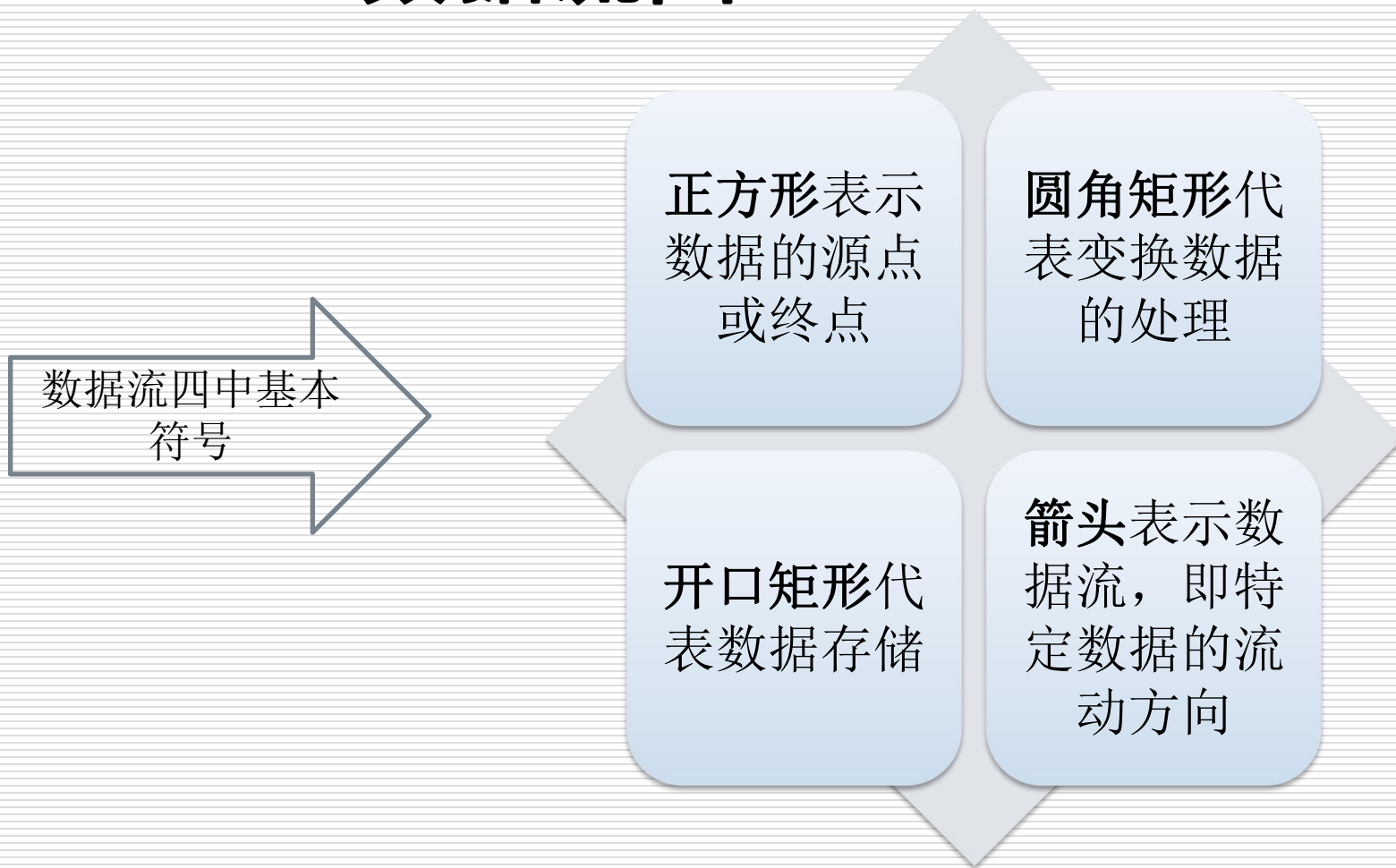


## 2.4 数据流图

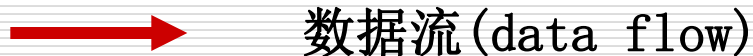
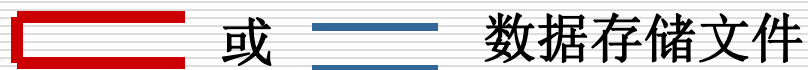
### DFD ---- Data Flow Diagram

- 一种图形化技术，它描绘信息流和数据从输入移动到输出的过程中所经受的变换。
- 在数据流图中没有任何具体的物理部件，它只是描绘数据在软件中流动和被处理的逻辑过程，是系统逻辑功能的图形表示。
- 设计数据流图时只需考虑系统必须完成的基本逻辑功能，完全不需要考虑怎样具体地实现这些功能，所以它也是今后进行软件设计的很好的出发点。

## 2.4 数据流图



## 2.4.1 数据流图四种基本符号

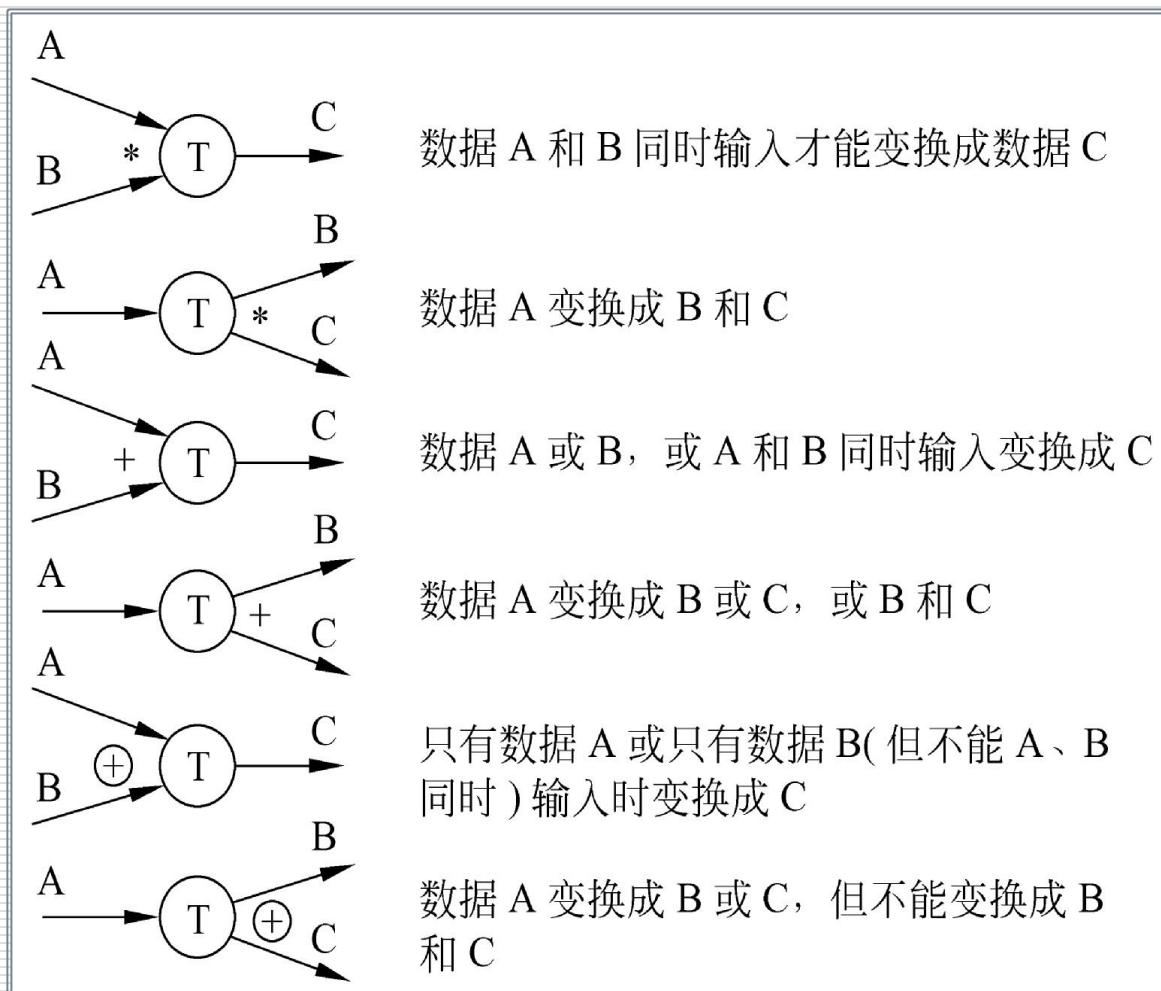


注意:

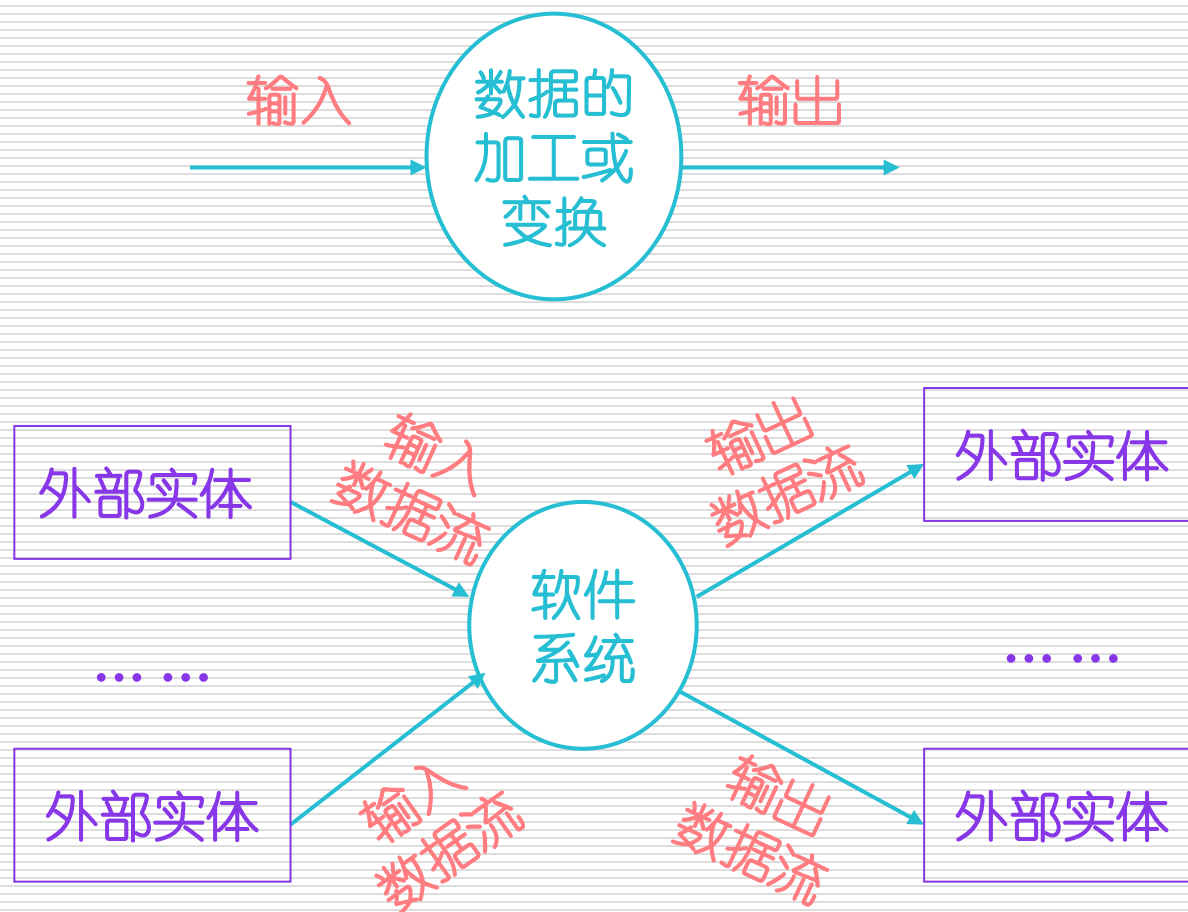
“**处理**”可表示：单个程序、一系列程序、程序的一个模块、人工处理过程等等；

“**数据存储**”可表示：一个文件、文件的一部分、数据库记录等等；数据流图忽略出错处理、打开文件、关闭文件。

# 数据流图几种附加符号

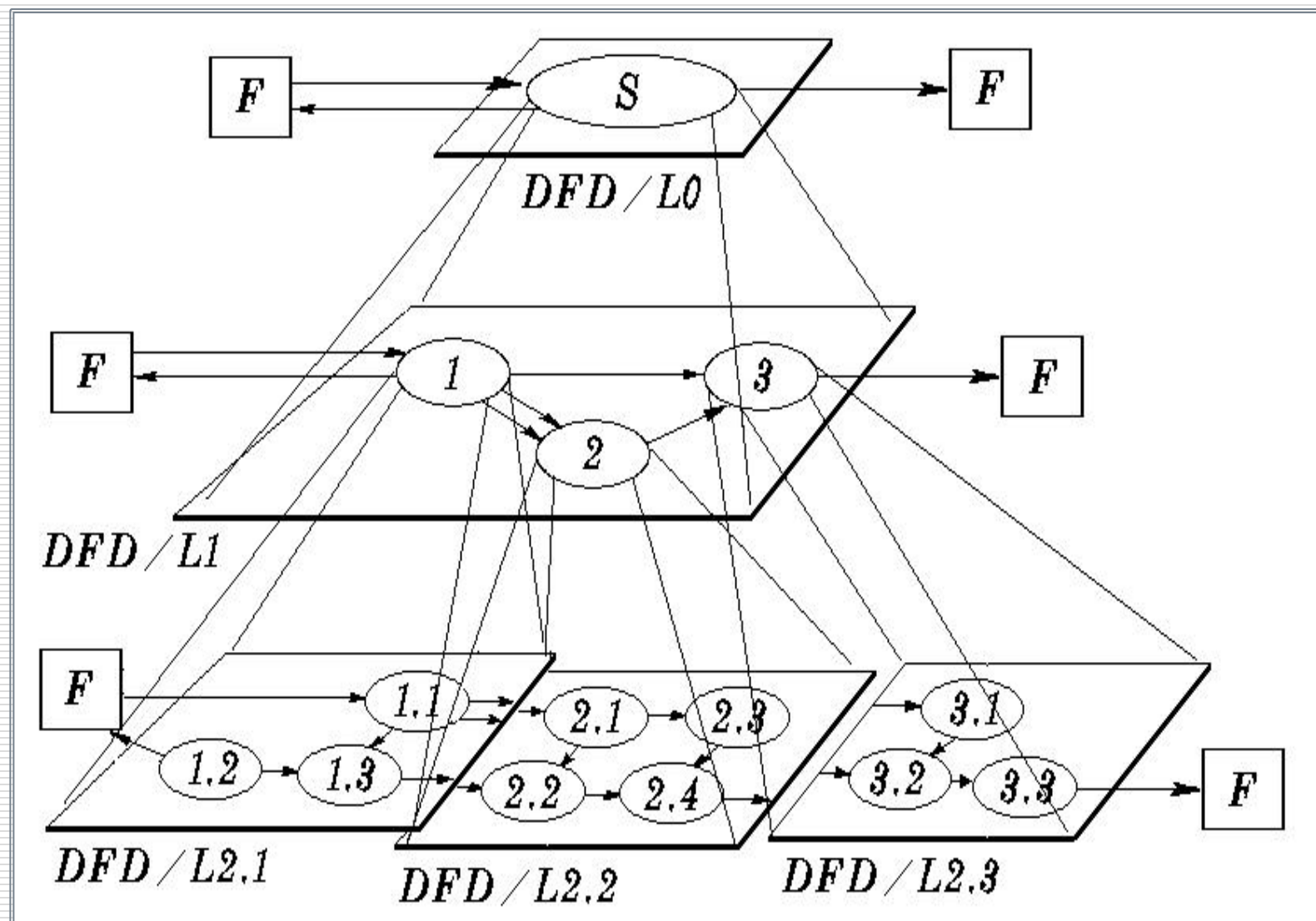


# 系统逻辑模型



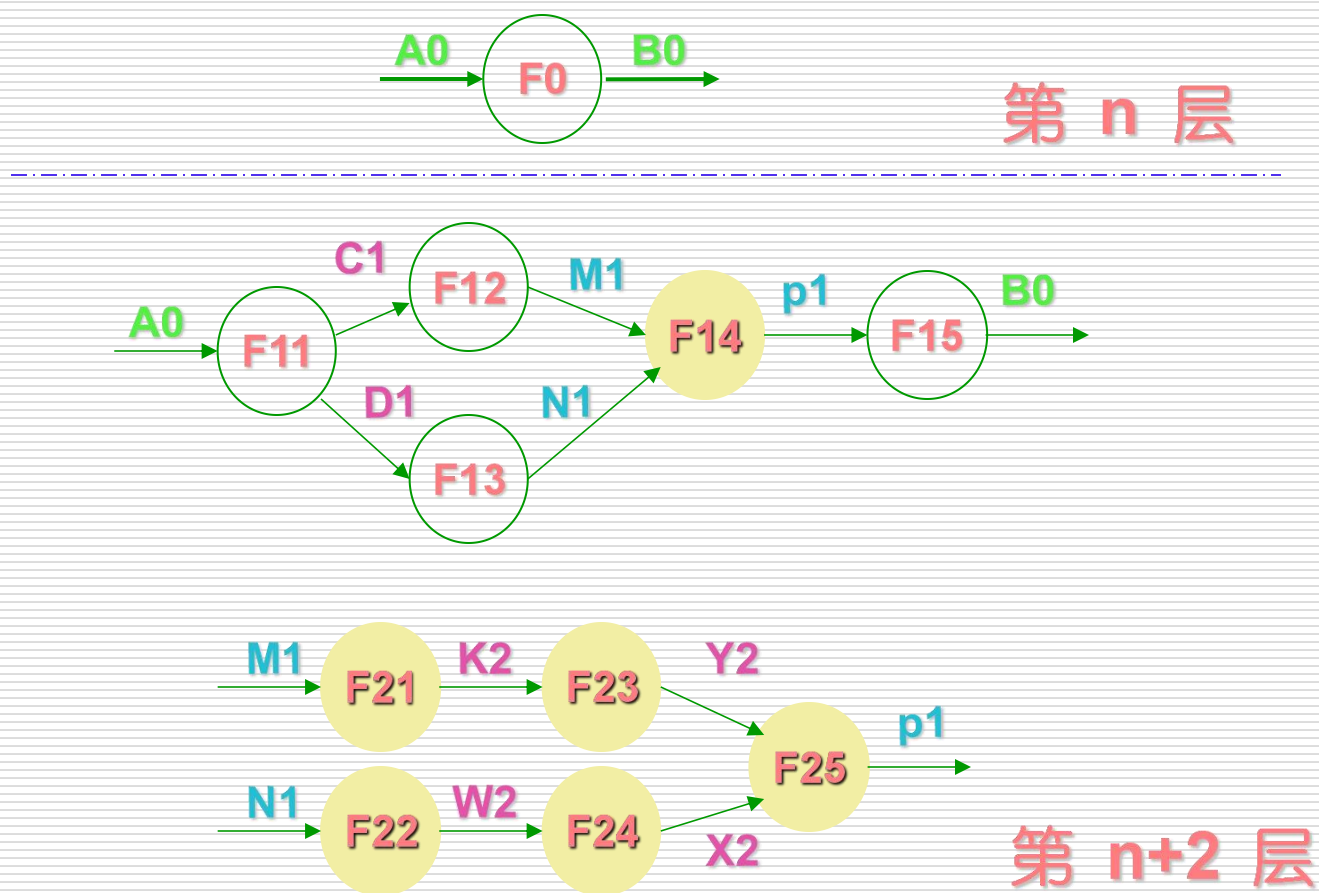
# 数据流图的层次结构

- 为了表达数据处理过程的数据加工情况，需要采用层次结构的数据流图。按照系统的层次结构进行逐步分解，并以分层的数据流图反映这种结构关系，能清楚地表达和容易理解整个系统。
- 在多层数据流图中，顶层流图仅包含一个加工，它代表被开发系统。它的输入流是该系统的输入数据，输出流是系统所输出数据。
- 底层流图是指其加工不需再做分解的数据流图，它处在最底层。
- 中间层流图则表示对其上层父图的细化。它的每一加工可能继续细化，形成子图。



分层的数据流图

# 分层的数据流图





## 2.4.2 绘制数据流图的例子

### —— 订货系统

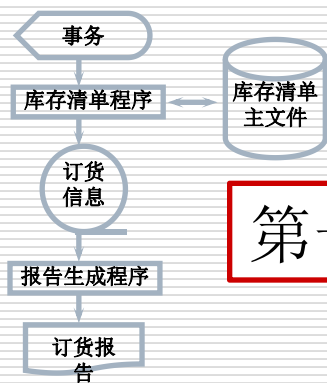
## 2.4 数据流图

## 订货系统

### 2.4.2 例子



假设一家工厂的采购部每天需要一张订货报表，报表按零件编号排序，表中列出所有需要再次订货的零件。对于每个需要再次订货的零件应该列出下述数据：零件编号，零件名称，订货数量，目前价格，主要供应者，次要供应者。零件入库或出库称为事务，通过放在仓库中的CRT终端把事务报告给订货系统。当某种零件的库存数量少于库存量临界值时就应该再次订货。



第一步可以从问题描述中提取数据流图的4种成分：

首先考虑数据的源点和终点，从上面对系统的描述可以知道

“采购部每天需要一张订货报表”，

“通过放在仓库中的CRT终端把事务报告给订货系统”，

描述中的数据是什么？

事务，订货报表

数据源点呢？终点呢？

仓库管理员，采购员

所以采购员接收来自系统的订货报表，而仓库管理员输入事务给系统。

思考：其他词汇？



## 第二步：再一次阅读问题描述，“采购部需要报表”

- 因此必须有一个用于产生报表的处理。
- 事务的后果是改变零件库存量，然而任何改变数据的操作都是处理，因此对事务进行的加工是另一个处理。
- 注意，在问题描述中并没有明显地提到需要对事务进行处理，但是通过分析可以看出这种需要。

产生报表  
处理事务

### 第三步：考虑数据流和数据存储

- 系统把订货报表送给采购部，因此订货报表是一个数据流；
- 事务需要从仓库送到系统中，显然事务是另一个数据流。
- 产生报表和处理事务这两个处理在时间上明显不匹配——每当有一个事务发生时立即处理它，然而每天只产生一次订货报表。因此，用来产生订货报表的数据必须存放一段时间，也就是应该有一个数据存储。

事务		订货报表	订货信息
		零件编号	(见订货报表)
零件编号		零件名称	库存清单
事务类型		订货数量	零件编号
数量		目前价格	库存量
		主要供应商	库存量临界值
		次要供应商	

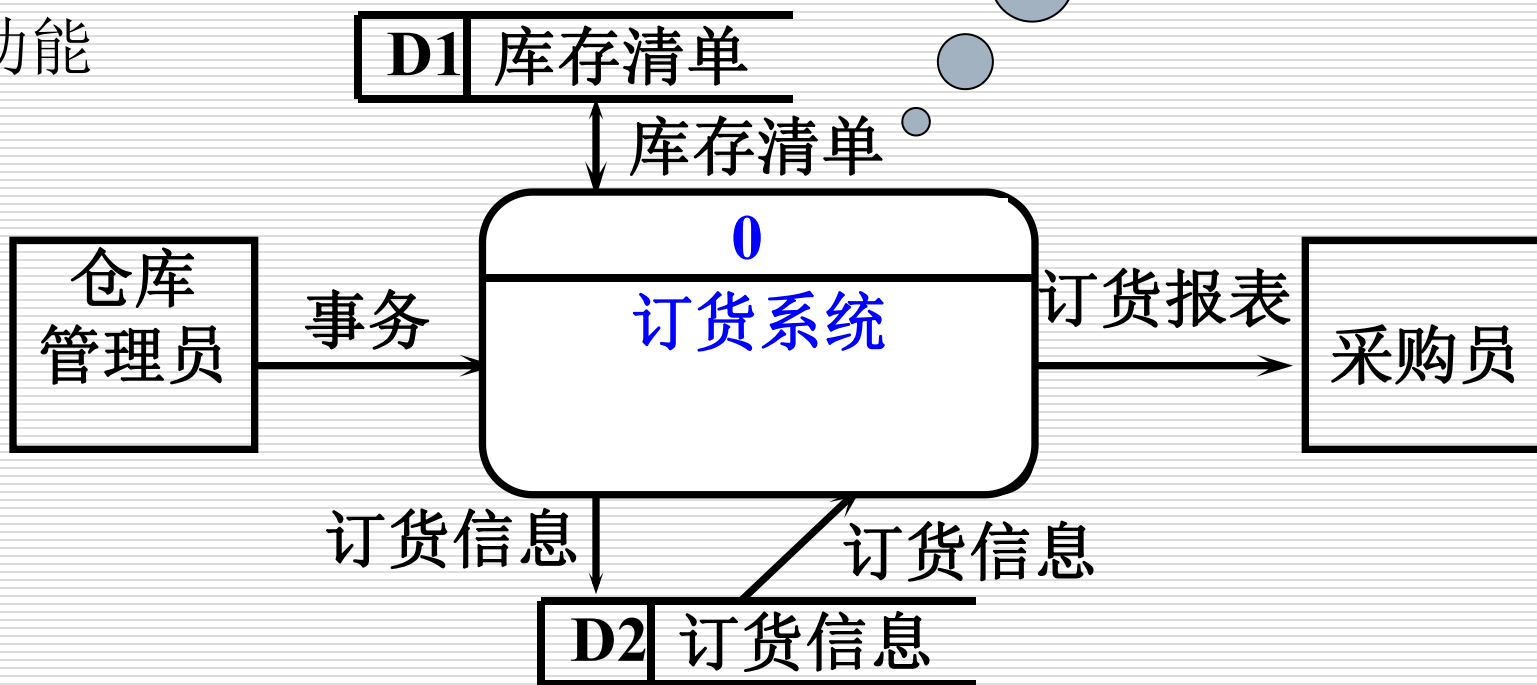
## 步骤一:

源点/终点	处理
采购员 仓库管理员	产生报表 处理事务
数据流	数据存储
事务 零件编号 事务类型 数量 订货报表 零件编号 零件名称 订货数量 目前价格 主要供应商 次要供应商	订货信息 (见订货报表) 库存清单 零件编号 库存量 库存量临界值


 分析结果

## 步骤二:

把基本系统  
模型细化，  
描绘系统的  
主要功能

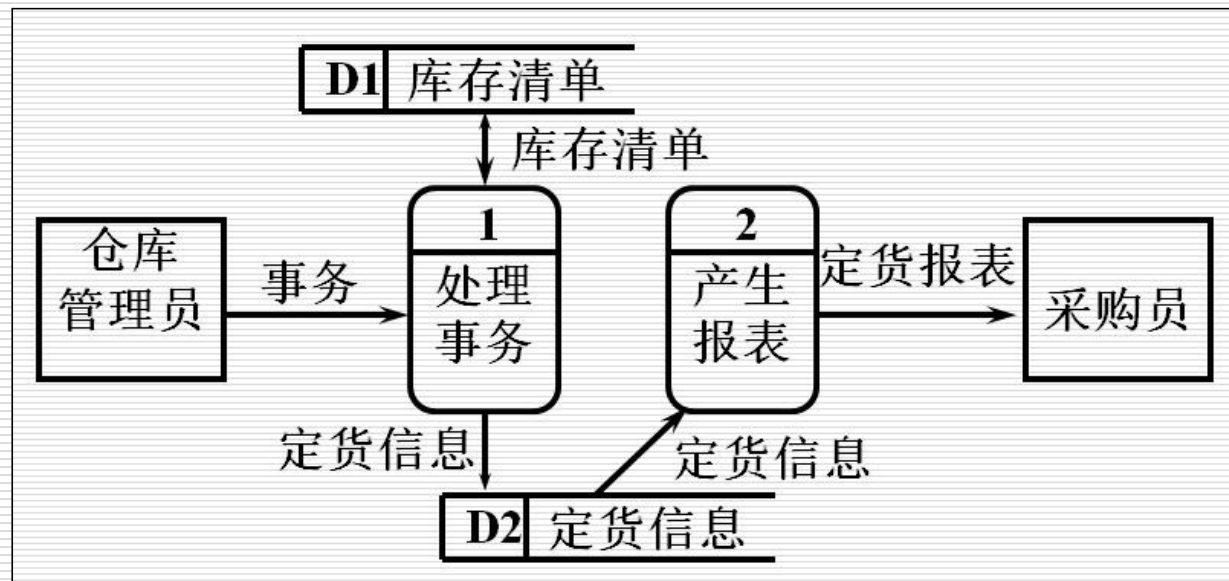


□ 上述数据流图所描述的功能够详细了吗？



### 步骤三：

对功能级数据流图中描绘的系统主要功能进一步细化



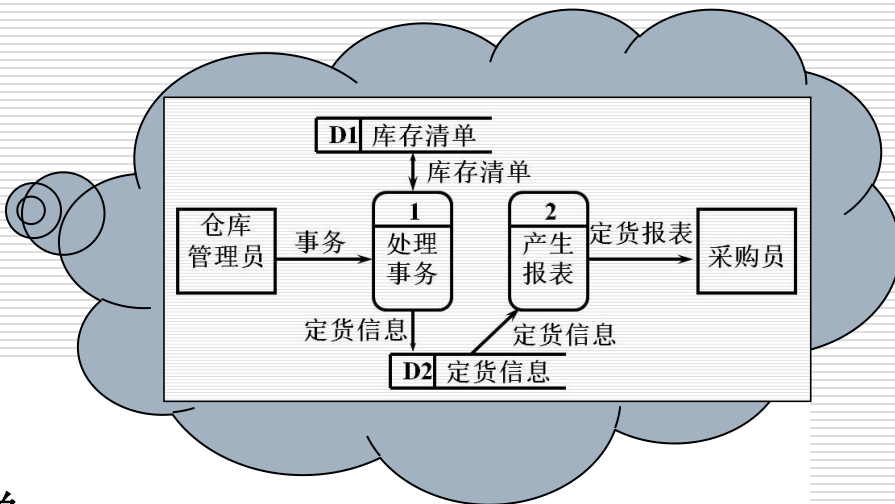
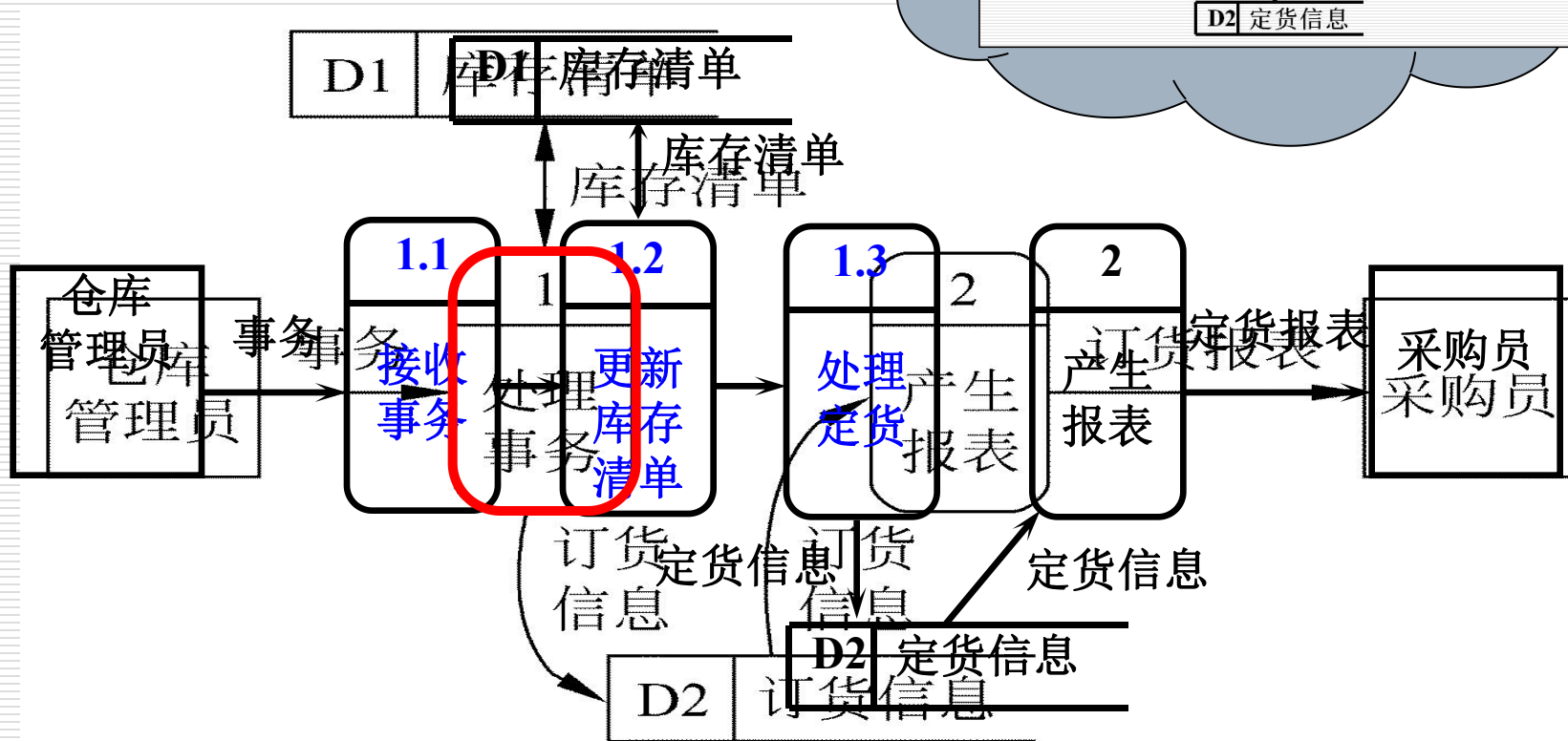
考虑通过系统的逻辑数据流（事务）：

- 当发生一个事务时必须首先**接收**它；
- 随后按照事务的内容**修改库存清单**；
- 最后如果更新后的库存量少于库存量临界值时，则应该再次订货，也就是需要**处理订货**信息。
- 因此，把“处理事务”这个功能分解为下述**3**个步骤，这在逻辑上是合理的：“**接收事务**”、“**更新库存清单**”和“**处理订货**”

# 步骤四：对功能级数据流图中描绘的系统主要功能进一步细化

1 处理事务

“1.1接收事务”  
“1.2更新库存清单”  
“1.3处理订货”



- 订货系统的功能级数据流图把处理事务的功能进行进一步分解，

思考：需要进一步分解“产生报表”这个功能吗？

订货报表中需要的数据在存储的订货信息中全都有，产生报表只不过是按一定顺序排列这些信息，再按一定格式打印出来。然而这些考虑纯属具体实现的**细节**，不应该在数据流图中表现。

同样道理，对“接收事务”或“更新库存清单”等功能也没有必要进一步细化。总之，当进一步分解将涉及如何**具体地实现**一个功能时就不应该再分解了。

●数据流图分层细化时必须保持**信息连续性**：当把一个处理分解为一系列处理时，分解前和分解后的输入输出数据流必须相同。

例如：“处理事务”这个处理框的输入输出数据流是“事务”、“库存清单”和“订货信息”，分解成“接收事务”、“更新库存清单”和“处理订货”3个处理之后，它们的输入输出数据流仍然是“事务”、“库存清单”和“订货信息”。

此外还应该注意处理进行编号的方法。处理1.1，1.2和1.3是更高层次的数据流图中处理1的组成元素。如果处理2被进一步分解，它的组成元素的编号将是2.1，2.2，...；如果把处理1.1进一步分解，则将得到编号为1.1.1，1.1.2，...的处理。



# 分层 DFD 图的优点

- 便于实现 --- 采用逐步细化的扩展方法，可避免一次引入过多的细节，有利于控制问题的复杂度；
- 便于使用 --- 用一组图代替一张总图，方便用户及软件开发人员阅读。

## 2.4.3 命名

### □1) 为数据流（或数据存储）命名

- **A.** 名字应该代表整个数据流（或数据存储）的内容；
- **B.** 不要使用空洞的、缺乏具体含义的名字（如“数据”、“输入”）；
- **C.** 如果为某个数据流（或数据存储）起名字时遇到困难，则很可能是因为没有对数据流图的分解不恰当造成的，应该试试重新分解数据流图；

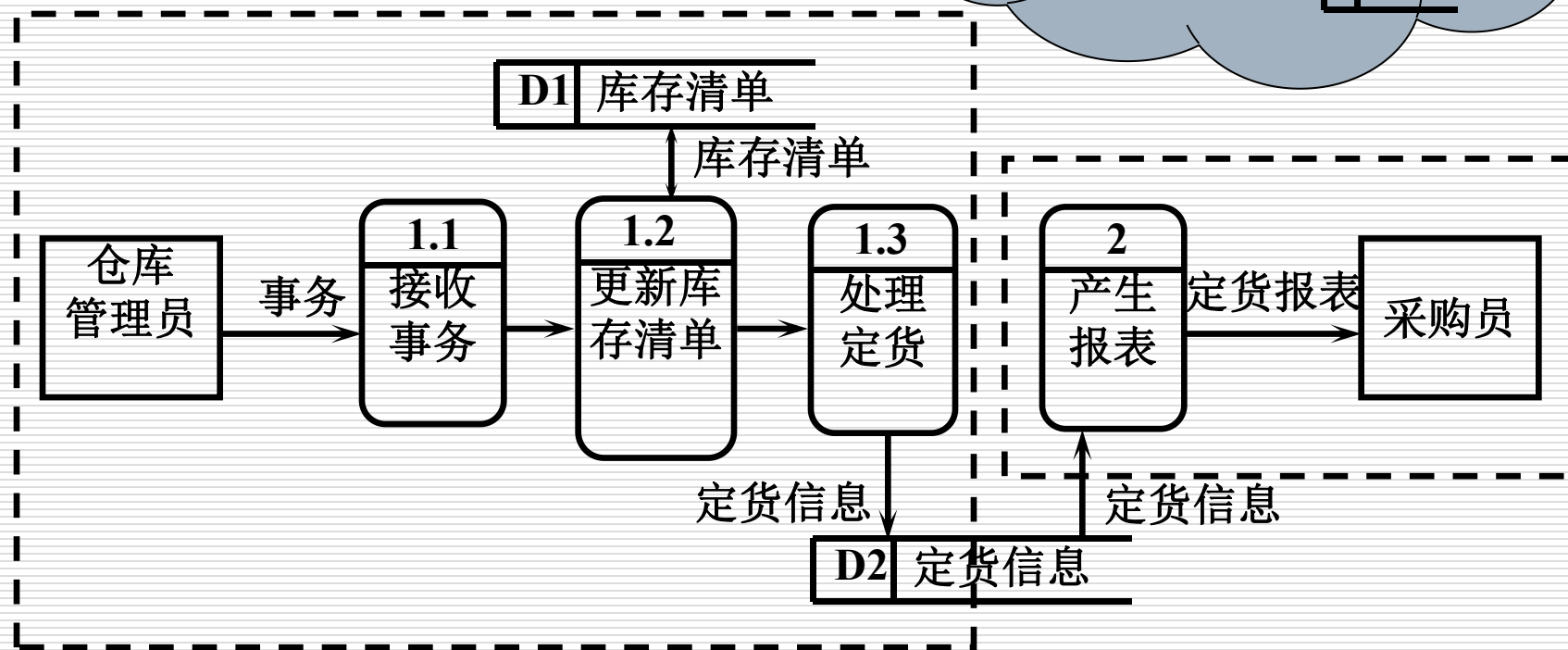
## □ 2) 为处理命名

- **A.** 通常先为数据流命名，然后再为与之相关联的处理命名；
  - **B.** 名字应该反映整个处理的功能；
  - **C.** 应该尽量避免空洞笼统的动词做名字，如“处理”、“加工”；
  - **D.** 通常用一个**动词**命名，如果必须用两个动词才能描述整个处理的功能，则可能要把这个处理分解成两个处理更恰当；
  - **E.** 如果在为某个处理命名时遇到困难，则很可能是发现了分解不当的情况，应考虑重新分解。
- 通常，为“数据源点/终点”命名时，采用它们在**问题域中习惯使用的名字**（如“仓库管理员”、“采购员”）

## 2.4.4 数据流图的用途

- **1)** 利用它作为交流信息的工具（分析员，有关人员）；
- **2)** 作为软件分析和设计的工具。（功能）

以联机方式更新库存清单

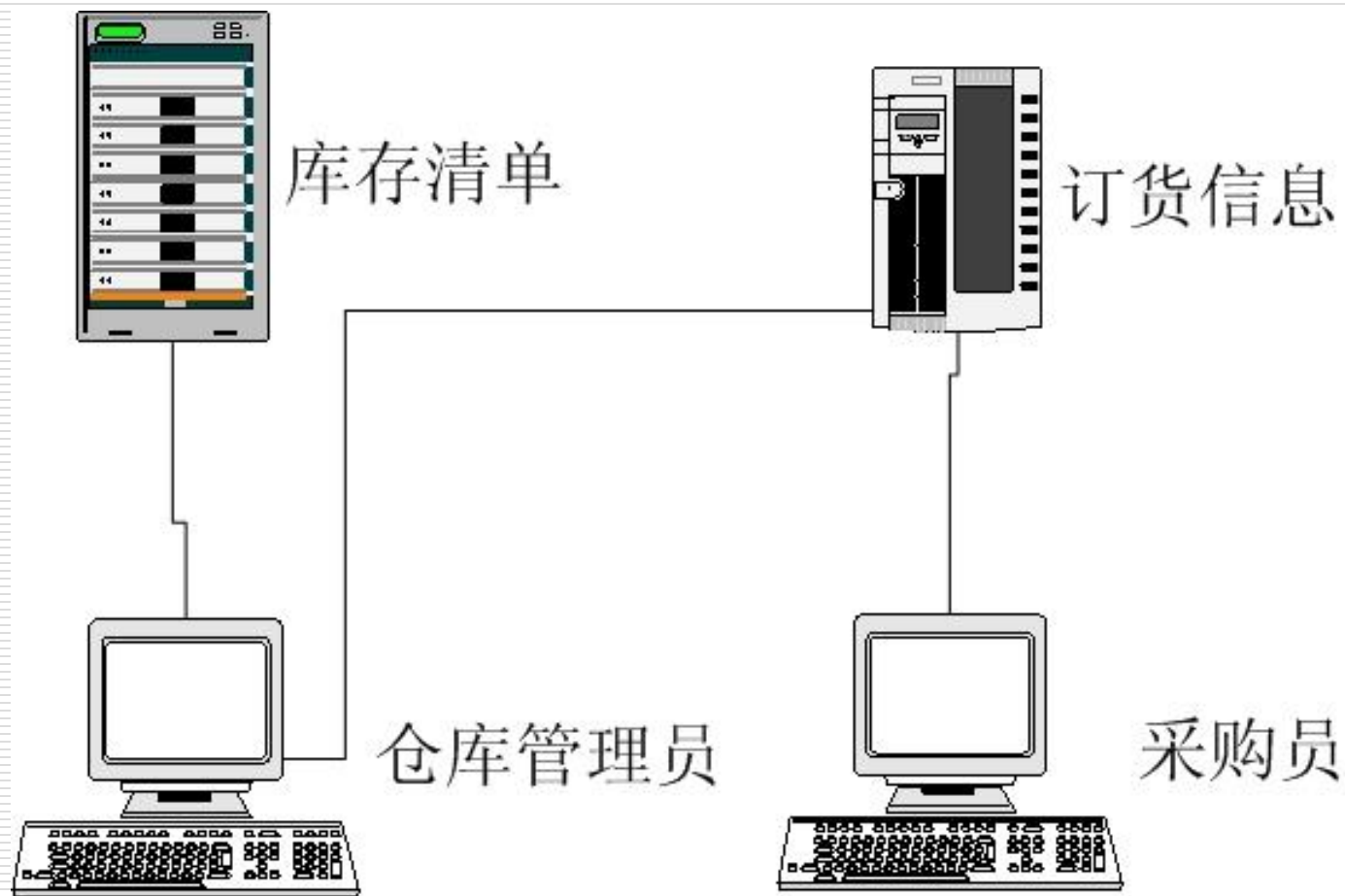


接收事务和处理

产生报表

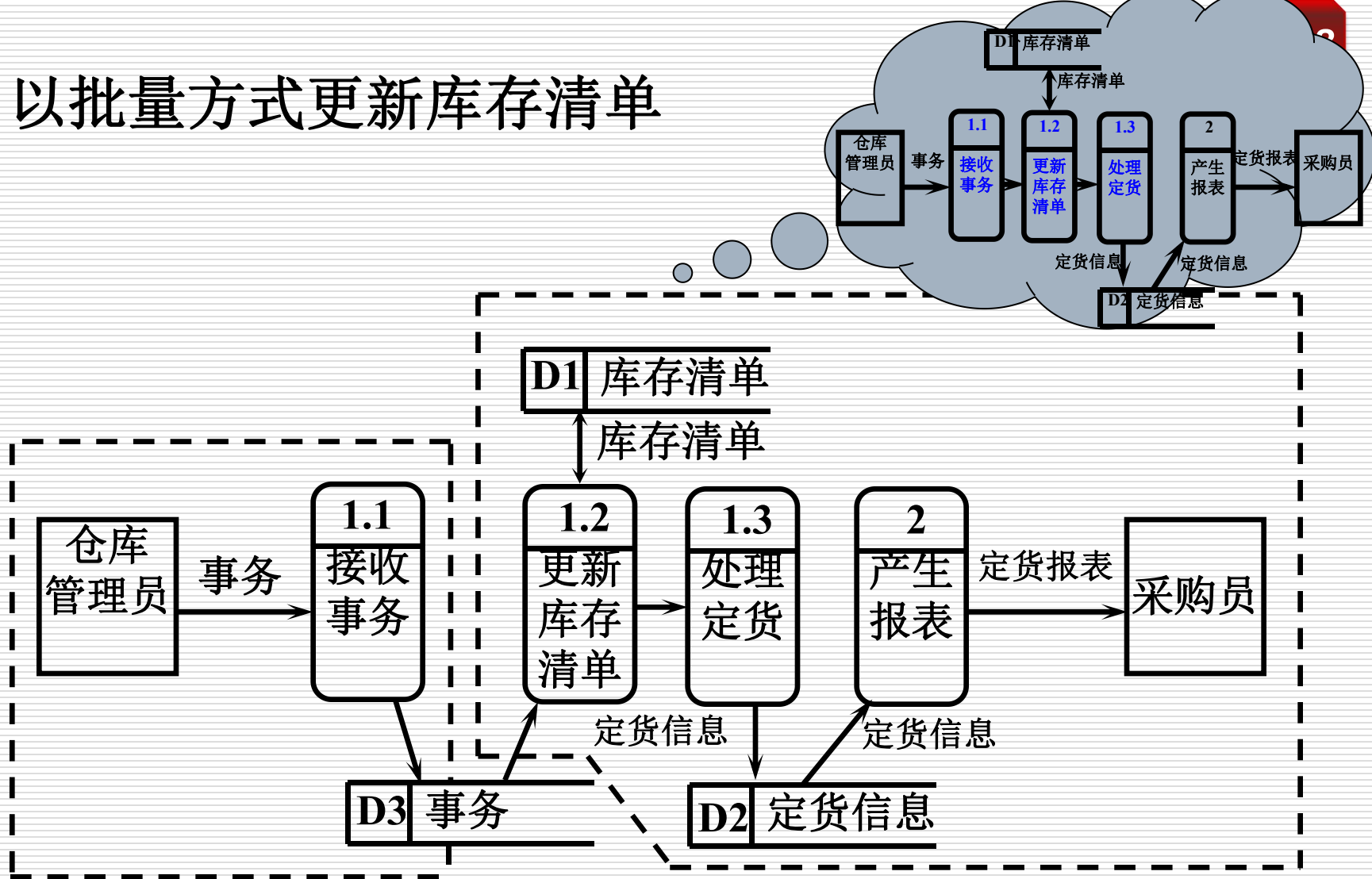
图3.8 这种自动化边界建议以联机方式更新库存清单





□ 图3.8 对应的物理实现硬件方案

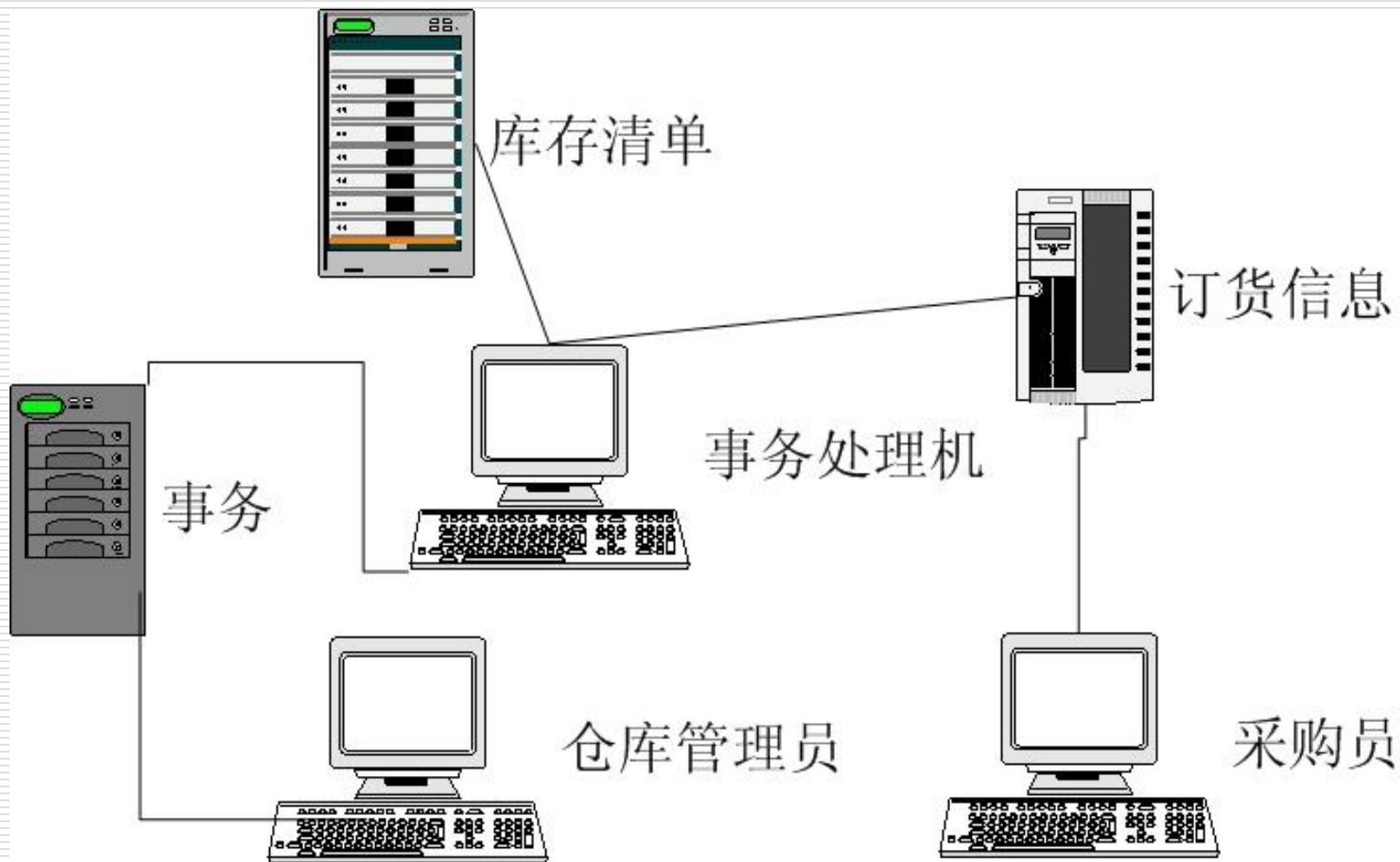
## □ 以批量方式更新库存清单



接收事务

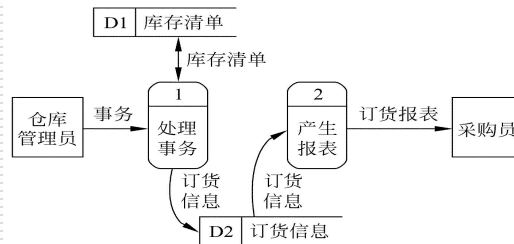
处理和产生报表

图3.9 这种自动化边界暗示以批量方式更新库存清单

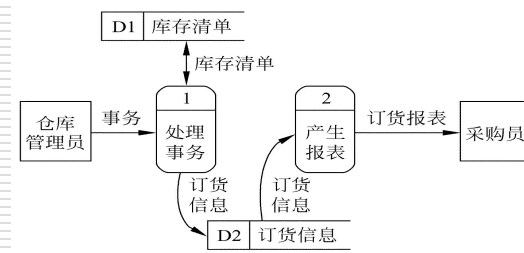


□ 图3.9 对应的物理实现硬件方案

## 2.5 数据字典

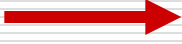

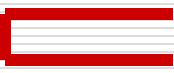
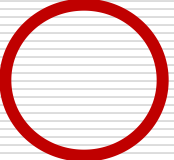


- 数据字典：对数据流图中包含的所有元素的定义的集合；
- 可行性研究阶段，数据流图与数据字典共同构成系统的逻辑模型。
- 数据字典最重要的用途是作为分析阶段的工具
  - 在数据字典中建立的一组严密一致的定义很有助于改进分析员和用户之间的通信，因此将消除许多可能的误解。
  - 对数据的这一系列严密一致的定义也有助于改进在不同的开发人员或不同的开发小组之间的通信。
  - 如果要求所有开发人员都根据公共的数据字典描述数据和设计模块，则能避免许多麻烦的接口问题。
- 数据字典是开发数据库的第一步，而且是非常有价值的一步。



## □ 2.5.1 数据字典的内容

### □ 数据字典应该对下列元素进行定义：

- **1) 数据流；**  如事务，库存清单，订货报表等
- **2) 数据元素（数据流分量）；**  如库存清单中的零件编号，数量等
- **3) 数据存储；**  如库存清单，订货信息等
- **4) 处理。**  如处理事务，产生报表等

## 2.5.2 定义数据的方法

- 1) 数据元素字典定义
- 其定义的基本内容有：
  - A. 数据元素编号、名称及其含义；
  - B. 数据类型和长度；
  - C. 合理取值；
  - D. 其他内容，如它与其它数据的逻辑关系等。

数据元素字典定义实例：学生成绩信息中的**考试成绩**

数据元素编号：DC001

数据元素名称：考试成绩

别名：成绩、分数

简述：学生考试成绩，分五个等级

类型/长度：两个字节，字符类型

取值/含义：优 [90-100]

良 [80-89]

中 [70-79]


及格 [60-69]

不及格 [0-59]

有关数据项或结构：学生成绩档案

有关处理逻辑：计算成绩


数据元素字典定义

- 2) 数据流字典定义 
- 其定义的基本内容有：
  - A. 数据流编号及名称；
  - B. 数据流来源；
  - C. 数据流去处；
  - D. 数据流的组成；
  - E. 流通量；
  - F. 峰值。

数据流字典定义实例：发往预定机票处理的**订票单**

数据流编号：DF001  
 数据流名称：订票单  
 简述：订票时填写的订票单  
 数据流来源：外部实体“乘客”  
 数据流去处：处理逻辑“预订机票”  
 数据流组成：订单编号  
                   日期  
                   乘客号  
                   航班号  
                   状态  
                   订单失效日期  
 流通量：每天300份  
 高峰值流通量：每天早上9：00，约160份

数据流字典定义

- 3) 数据存储字典定义 
- 其定义的基本内容有：
  - A. 数据存储编号及名称；
  - B. 数据存储的组成；
  - C. 其它要求。



## 数据处理字典定义实例：

- 4) 数据处理字典定义
- 其定义的基本内容有：
- A. 数据处理编号及名称；
- B. 简单描述；
- C. 输入/输出；
- D. 功能描述；
- E. 有关数据存储。

**数据处理编号：DP001**

**数据处理名称：编辑订票**

**简述：**接收从终端录入的订票单，检验是否正确

**输入：**乘客订单，**来源：**外部实体“乘客”

**输出：**1.合格订单，**去处：**处理逻辑“确定订票”

2.不及格订单，**去处：**外部实体“乘客”

**功能描述：**……（略）

**有关数据存储：**订票记录

数据处理字典定义

# 组成数据项的表示方法

符号	含 义	例 子
=	被定义为	
+	与	$x=a+b$ , 则表示 $x$ 由 $a$ 和 $b$ 组成
[ ]	或	$x=[a,b]$ , $x=[a b]$ 则表示 $x$ 由 $a$ 或由 $b$ 组成
{ }	重复	$x=\{a\}$ , 则表示 $x$ 由 0个或多个 $a$ 组成
$m\{ \}n$	重复	$x=3\{a\}8$ , 则表示 $x$ 中至少出现3次 $a$ ,最多出现8次
( )	可选	$x=(a)$ , 则表示 $a$ 在 $x$ 中出现, 也可不出现 (可选)
..	连结符	$x=1..9$ , 则表示 $x$ 为1到9中的数字
*...*	注释符	表示在两个 * 之间的内容为词条的注释

# 例：存折 (数据存储)

户名:.....

所号:xxx

帐号:xxxxxxxx

开户日:xxxx年xx月xx日

日期(年月日)	摘要	支出	存入	余额	操作	复核
XXXX XX XX	1~4					

性质:x

印密:0

存取行  
1~20

存折 = 户名+所号+帐号+开户日+性质+(印密)+1{存取行}20

户名 = 2{字母}24

所号 = “001” .. “999”

帐号 = “00000001” .. “99999999”

开户日 = 年+月+日

性质 = “1” .. “6” 注：“1”表示普通户，...“6”表示工资户等

印密 = “0” 注：印密在存折上不显示

存取行 = 日期+（摘要）+支出+存入+余额+操作+复核

日期 = 年+月+日

年 = “1900” .. “3000” 月 = “01” .. “12” 日 = “01” .. “31”

摘要 = 1{字母}4

支出 = 金额

金额 = “00000000.01” .. “999999999.99” ... ..

## 2.5.3 数据字典的用途

- **1.** 作为分析阶段的重要工具;
- **2.** 数据元素的控制信息非常有用;
- **3.** 有助于开发数据库。

## 2.5.4 数据字典的实现

- CASE结构化分析与设计工具（大型软件）
- 卡片形式/excel or record in file（小型软件）

名字:  
别名:  
描述:  
定义:  
位置:

名字:订货报表  
别名:订货信息  
描述:每天一次送给采购员的需要订货的零件表  
定义:订货报表 = 零件编号 + 零件名称 + 订货数量 + 目前价格 + 主要供应者 + 次要供应者  
位置:输出到打印机

■ ■ ■ ■ ■ ■

## 2.6 成本/效益分析

### 2.6.1 成本估计

1. 代码行技术
2. 任务分解技术
3. 自动估计成本技术

#### □ 1) 代码行技术

- 根据经验和历史数据估计实现一个功能需要的源程序行数，用每行代码的平均成本乘以行数就可以确定软件的成本。
- 每行代码的平均成本主要取决于软件的复杂程度和工资水平。
- 代码行技术是比较简单的定量估算方法。当有以往开发类似工程的历史数据可供参考时，这个方法是非常有效的。

□ 软件成本 = 每行代码的平均成本 × 估计的源代码总行数

## 2) 任务分解技术

软件开发项目分解为若干个相对独立的任务，分别估计每个单独任务的成本：

单独任务成本 = 任务所需人力估计值 × 每人每月平均工资；

软件开发项目总成本估计 = 各个单独任务成本估计值之和。

□ 常用的办法是按开发阶段划分任务，典型环境下各个开发阶段需要使用的人力百分比大致如下：

任务	人力（%）
可行性研究	5
需求分析	10
设计	25
编码与单元测试	20
综合测试	40
总计	100

### □ 3) 自动估计成本技术

□ 采用自动估计成本的软件工具估计。

# 软件开发成本估算的经验模型：

多种因素  
影响

## □1) Putnam 模型

□ 1978年Putnam提出的，一种动态多变量模型：

The diagram shows the Putnam model equation  $L = C_k \cdot K^{1/3} \cdot t_d^{4/3}$  with lines connecting each variable to its definition.  $L$  is defined as '源代码行数 (以LOC计)'.  $K$  is defined as '软件开发与维护在内的整个生存期所花费的工作量 (以人年计)'.  $C_k$  is defined as '技术状态常数'.  $t_d$  is defined as '开发持续时间 (以年计)'.

$L$  是源代码行数 (以LOC计)

$K$  是软件开发与维护在内的整个生存期所花费的工作量 (以人年计)

$$L = C_k \cdot K^{1/3} \cdot t_d^{4/3}$$

$C_k$  是技术状态常数。

$t_d$  是开发持续时间 (以年计)



- **Ck**为技术状态常数，它反映“妨碍开发进展的限制”，取值因开发环境而异，见下表：

Ck的典型值	开发环境	开发环境举例
2000	差	没有系统的开发方法，缺乏文档和复审
8000	好	有合适的系统的开发方法，有充分的文档和复审
11000	优	有自动的开发工具和技术

## □2) COCOMO模型(constructive cost model)

□ 这是由TRW公司开发，Boehm提出的结构化成本估算模型，是一种精确的、易于使用的成本估算方法。

□ 基本COCOMO模型估算工作量和进度的公式如下：

□ 工 作 量：  $MM = r \times (KDSI)^c$  （人月）

□ 开发时间：  $TDKV = a \times (MM)^b$  （月）

□ DSI：源指令条数，不包括注释， $1KDSI = 1000DSI$

□ MM：开发工作量（以人月计）， $1MM = 19 \text{ 人日} = 152 \text{ 人时} = 1/12 \text{ 人年}$

□ 经验常数  $r, c, a, b$  取决于项目的总体类型

□ COCOMO模型中，考虑开发环境，软件开发项目的类型可以分为3种：

■ 1) 组织型(organic)

■ 相对较小、较简单的软件项目。开发人员对开发目标理解比较充分，与软件系统相关的工作经验丰富，对软件的使用环境很熟悉，受硬件的约束较小，程序的规模不是很大（<50000行）

■ 2) 嵌入型(embedded)

■ 要求在紧密联系的硬件、软件 and 操作的限制条件下运行，通常与某种复杂的硬件设备紧密结合在一起。对接口，数据结构，算法的要求高。软件规模任意。

■ 如大而复杂的事务处理系统，大型/超大型操作系统，航天用控制系统，大型指挥系统等。

■ 3) 半独立型(semidetached)

■ 介于上述两种软件之间。规模和复杂度都属于中等或更高。最大可达30万行。

□ **COCOMO**模型按其详细程度可以分为三级：

□ 1) **基本COCOMO模型**

□ 是一个静态单变量模型，它用一个以已估算出来的源代码行数(LOC)为自变量的经验函数计算软件开发工作量。

□ 通过统计63个历史项目的历史数据，得到如下计算公式：

总体类型	工作量	所需开发时间
组织型	$MM = 2.4 \times (KDSI)^{1.05}$	$TDKV = 2.5 \times (MM)^{0.38}$
半独立型	$MM = 3.0 \times (KDSI)^{1.12}$	$TDKV = 2.5 \times (MM)^{0.35}$
嵌入型	$MM = 3.0 \times (KDSI)^{1.20}$	$TDKV = 2.5 \times (MM)^{0.32}$

## □2) 中级COCOMO模型

□ 在基本COCOMO模型的基础上，再用涉及产品、硬件、人员、项目等方面的影响因素调整工作量的估算。

## □3) 详细COCOMO模型

□ 包括中级COCOMO模型的所有特性，但更进一步考虑了软件工程中每一步骤（如分析、设计）的影响。

## 2.6.2 成本/效益分析的方法

- 成本/效益分析的第一步是估计开发成本、运行费用和新系统将带来的经济效益，然后从经济角度判断这个系统是否值得投资。
- 运行费用取决于系统的操作费用(操作员人数，工作时间，消耗的物资等)和维护费用。
- 系统的经济效益等于因使用新系统而增加的收入加上使用新系统可以节省的运行费用。

### 注意：

- 总的效益和生命周期的长度有关，所以应该合理地估计软件的寿命。
- 投资是现在进行的，效益是将来获得的，不能简单地比较成本和效益，应该考虑货币的时间价值。

# 成本/效益分析使用的几个概念

## 1. 货币的时间价值

通常用利率表示货币的时间价值。设年利率为*i*，现已存入*P*元，则*n*年后可得钱数为

$$F=P(1+i)^n$$

这就是*P*元钱在*n*年后的价值。反之，若*n*年后能收入*F*元，那么这些钱现在的价值是

$$P=F/(1+i)^n$$

## 2. 投资回收期

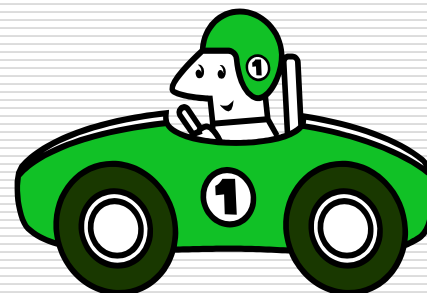
投资回收期是衡量一个开发工程价值的经济指标。投资回收期就是积累的经济效益等于最初的投资所需要的时间。投资回收期越短，就能越快获得利润。

# 成本/效益分析使用的几个概念：

## 3. 纯收入

工程的纯收入是衡量工程价值的另一项经济指标。纯收入就是在整个生存周期之内系统的累计经济效益（折合成现在值）与投资之差。

如果纯收入小于零，那么显然这项工程不值得投资。只有当纯收入大于零，才能考虑投资。





## □4 投资回收率

$$P = F_1 / (1 + j) + F_2 / (1 + j)^2 + \cdots + F_n / (1 + j)^n$$

- 其中： $P$ 是现在的投资额；
- $F_i$ 是第 $i$ 年年底的效益（ $i=1,2,3,\dots,n$ ）；
- $n$ 是系统的使用寿命（一般假设 $n=5$ ）；
- $j$ 是投资回收率。
- 上述修改系统的工程的投资回收率是41%-42%

- 例：修改一个已有的库存管理系统，使它能在每天送给采购员一份订货报表。
- 修改已有的库存清单程序并且编写产生报表的程序，估计共需**5000**元；系统修改后能及时订货，这将消除零件短缺问题，估计因此每年可以节省**2500**元，**5**年共可节省**12500**元。
- 但是，不能简单地把**5000**元和**12500**元相比较，因为前者是现在投资的钱，后者是若干年以后节省的钱。
- 假定年利率为**12%**，利用上面计算货币现在价值的公式可以算出修改库存清单系统后每年预计节省的钱的现在价值，如下表所示。求项目的投资回收期。

□ **表1：将来的收入折算成现在值**

年	将来值 (元)	$(1+0.12)^n$	现在值 (元)	累计的现在 值 (元)
1	2500	1.12	2232.14	2232.14
2	2500	1.25	1992.98	4225.12
3	2500	1.40	1779.45	6004.57
4	2500	1.57	1588.80	7593.37
5	2500	1.76	1418.57	9011.94

## □2) 投资回收期

□ 第一、第二年回收：4225元

□ 第三年用于回收投资要：

□  $(5000 - 4225) / 1779 = 0.44$ 年

□ 总的投资回收期 = 2.44年

## 3) 纯收入

$$9011.94 - 5000 = 4011.94 \text{ (元)}$$

## 2.7 可行性研究的文档

- 规格说明书要按照如下原则进行：
  - **(1)** 从实现中抽出功能度。
  - **(2)** 用面向过程的系统规格说明语言。
  - **(3)** 规格说明要围绕整个系统，软件是其组成部分。
  - **(4)** 规格说明书是一个可以认知的模型。
  - **(5)** 必须是局部化的和松散耦合的。
  - **(6)** 必须围绕系统的操作环境。
  - **(7)** 必须是可以操作的。
  - **(8)** 允许系统的规格说明书的不完整和可扩充。

# 可行性研究报告的编写

GB 8567-88 《 计算机软件产品开发文件编制指南 》

## 1 引言

1.1 编写目的

1.2 背景

1.3 定义

1.4 参考资料

## 2 可行性研究的前提

2.1 要求

2.2 目标

2.3 条件、假定和限制

2.4 进行可行性研究的方法

2.5 评价尺度

# 可行性研究报告的编写

## 3 对现有系统的分析

- 3.1 数据流程和处理流程
- 3.2 工作负荷
- 3.3 费用开支
- 3.4 人员
- 3.5 设备
- 3.6 局限性

## 4 所建议的系统

- 4.1 对所建议系统的说明
- 4.2 数据流程和处理流程
- 4.3 改进之处
- 4.4 影响
- 4.5 局限性
- 4.6 技术条件方面的可行性

# 可行性研究报告的编写

## 5 可选择的其它系统方案

5.1 可选择的其它系统1

5.2 可选择的其它系统2

.....

## 7 社会条件方面的可行性

7.1 法律方面的可行性

7.2 使用方面的可行性

## 6 投资及收益分析

6.1 支出

6.2 收益

6.3 收益/投资比

6.4 投资回收周期

6.5 敏感性分析



## 2.8 项目开发计划

- **2.8.1 方案选择**
- 系统分析任务完成后，系统分析员开始研究问题求解方案。
- 由于系统开发成本又可划分为研究成本、设计成本、设备成本、程序编码成本、测试和评审成本、系统运行和维护成本、系统退役成本等，因此在开发系统所用总成本不变的情况下，由于系统开发各阶段所用成本分配方案的不同，也会对系统的功能和性能产生相当大的影响。

## 2.8.2 制定项目开发计划

- 经过可行性研究后，就得到一个项目是否值得开发的结论。如果可行，则接下来应制定**项目开发计划**。分析员应当进一步为推荐的系统编写一份开发计划。
- 软件项目开发计划是软件工程中的一种管理性文档。主要是对开发的软件项目的**费用、时间、进度、人员组织、硬件设备的配置、软件开发环境和运行环境的配置等**进行说明和规划，是项目管理人员对项目进行管理的**依据**，据此对项目的费用、进度和资源进行**控制和管理**。

- 项目开发计划是一个管理性文档，它的主要内容如下：
- （1）**项目概述**：说明项目的各项主要工作；说明软件的功能、性能；为完成项目应具备的条件；用户及合同承包者承担的工作、完成期限及其他条件限制；应交付的程序名称，所使用的语言及存储形式；应依附的文档。
- （2）**实施计划**：说明任务的划分，各项任务的责任人；说明项目开发进度，按阶段应完成的任务，用图表说明每项任务的开始时间和完成时间；说明项目的预算，各阶段的费用支出预算。
- （3）**人员组织及分工**：说明开发该项目所需人员的类型、组成结构和数量等。
- （4）**交付期限**：说明项目最后完工交付的日期。

# 小结

2.1 可行性研究的任务: **技术, 经济, 操作, 社会, 抉择**

---

2.2 可行性研究过程

2.3 系统流程图: 数据在系统(人, 软, 硬) 各部件之间流动

2.4 数据流图: 源点或终点, 处理, 数据存储和数据流

2.5 数据字典: 流, 元素, 存储, 处理

2.6 成本/效益分析: 估计, ROI, 货币的时间价值