



第4章 结构化开发方法

掌握并能正确运用开发方法，具有事半功倍的作用。

概念：软件开发方法

--软件开发过程所遵循的办法和步骤。

软件开发活动的目的是有效地得到一个运行的系统及其支持文档，并且满足有关的质量要求。

软件开发方法学

--指的是规则、方法和工具的集成，既支持开发也支持以后的演化过程（交付运行后，系统还会变化；或者为了改错，或为了功能的增减）。

From: 《计算机科学技术百科全书》(第二版)



北京大学



结构化方法

--一种特定的软件开发方法学（由Edward Yourdon, Tom Demarco等人提出）

一种系统化的软件开发方法，包括：

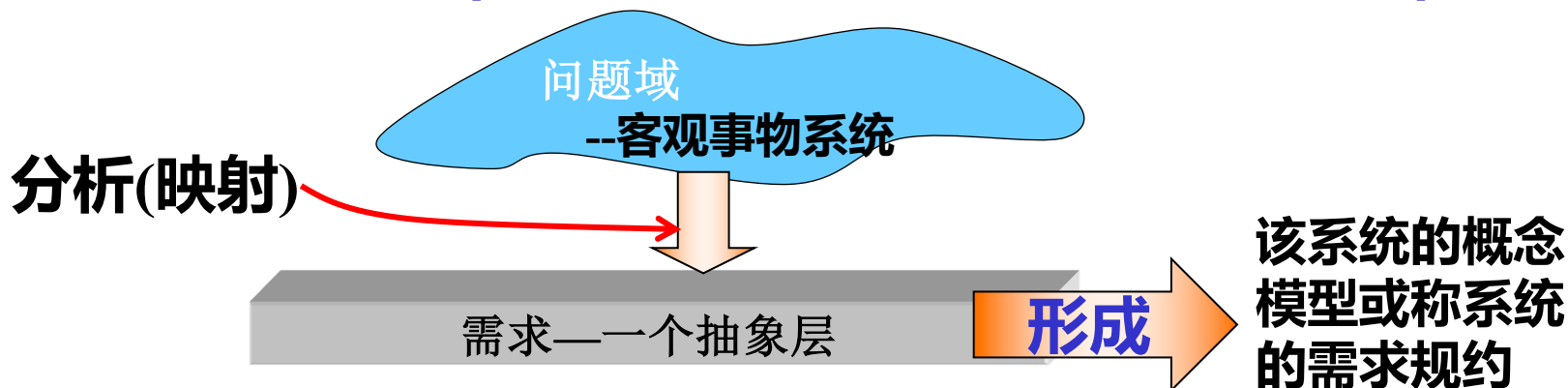
- 结构化分析方法**
- 结构化设计方法**
- 结构化程序设计方法**



北京大学

4.1 结构化分析

就软件需求分析而言，即：系统化地使用问题域术语，给出该问题的模型（即“系统必须做什么”的一个估算）。



1. 需求分析的目标

对需求陈述进行分析，解决其中的歧义、不一致等问题，以系统化的形式表达用户的需求，即给出问题的形式化或半形式化的描述（称为**系统的概念模型**，或**系统的需求规约**或**需求规格说明**）。作为开发人员和客户间技术契约的基础，并作为而后开发活动的一个基本输入。



北京大学



2. 实现软件需求分析的目标对方法学的需求

(1) **提供一组术语(符号)**，指导需求抽象中需要关注的主要方面，并用于表达分析中所使用的信息。

这些术语形成一个特定的抽象层，即需求层。

(2) 依据这些术语所形成的“空间”，**给出表达模型的工具**，支持表达系统功能形态。

(3) **给出过程指导**，以支持系统化地使用相关信息建造系统模型。





3. 基本术语

一个抽象层是由一组确定的术语定义的，为支持需求分析中有关要使用的那些信息的表达，结构化分析方法给出了以下五个术语 / 符号：

- ◆数据流：——→：数据的流动
- ◆加工：●：对数据进行变换的单元
- ◆数据存储：——：数据的静态结构
- ◆数据源：■：数据流的起点，系统之外的实体
- ◆数据潭：■：数据流的归宿地，系统之外的实体





其中：

- ① 数据流、数据存储--支持数据抽象，
加工--支持过程/功能的抽象，用于表达系统内涵。**
- ② 数据源、数据潭—支持系统边界抽象，用于表达系统外延。**
- ③ 是完备的。**

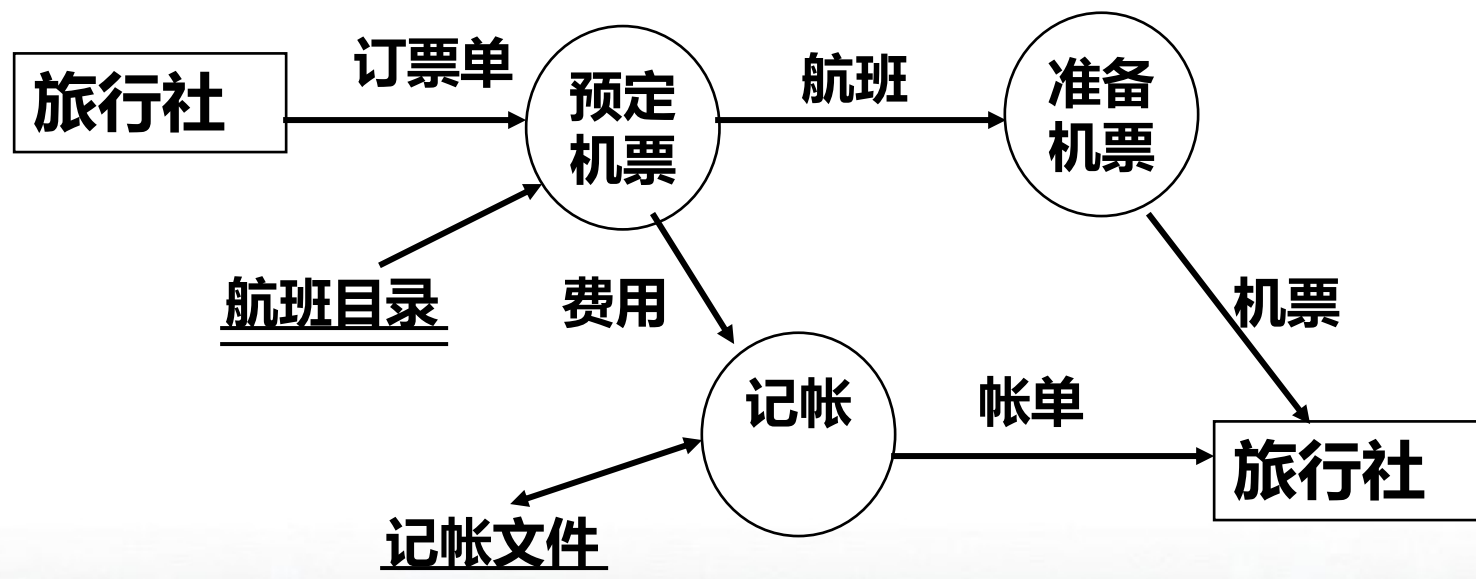


需求分析的首要任务是建立系统功能模型！

4. 模型表达工具

(1) 数据流图 (DFD图) —— 表达系统功能模型的工具

是一种描述数据变换的图形工具，它包含的元素可以是数据流、数据存储、加工、数据源和数据潭等。



一个飞机票预订系统的数据流图



北京大学



(2) 数据字典——定义数据流和数据存储

用于定义数据流和数据存储的结构，并给出构成所给出的数据流和数据存储的各数据项的基本数据类型。

引入：一些逻辑操作符——用于定义数据结构

操作符	含义描述
=	等价于（定义为）
+	与（顺序结构）
{ }	重复（循环结构）
[]	或（选择结构）
()	任选
m..n	界域





例如：

数据字典：

①、数据流：

销售的商品=商品名+商品编号+单价+数量+销售时间

现金额 = 余额 = 日销售额=非负实数

查询要求=[商品编号|日期]

查询要求1=商品编号

查询要求2=日期

销售情况=商品名+商品编号+金额

②、数据存贮：

销售文件={销售的商品}

③、数据项（数据流及数据存储的组成成分）

金额=非负实数

.....



北京大学



(3) 判定表或判定树等——定义加工小说明

描述加工“做什么”，即加工逻辑，也包括其它一些与加工有关的信息，如执行条件、优先级、执行频率、出错处理等。

① 结构化自然语言

---适用于加工的输入数据和输出数据之间的逻辑关系比较简单的加工描述

例如：if 20<订票量
then 订票折扣为10%
else 订票折扣为5%





②判定表:

适用于加工的输入数据和输出数据之间的逻辑关系比较复杂的加工描述。

判断表	I 条件类别	II 条件组合
	III 操作	IV 操作执行

例如:

考试总分	≥ 620	≥ 620	< 620
单科成绩	有满分	有不及格	有满分
发升级通知书	y	y	n
发留级通知书	n	n	y
发重修通知书	n	y	n



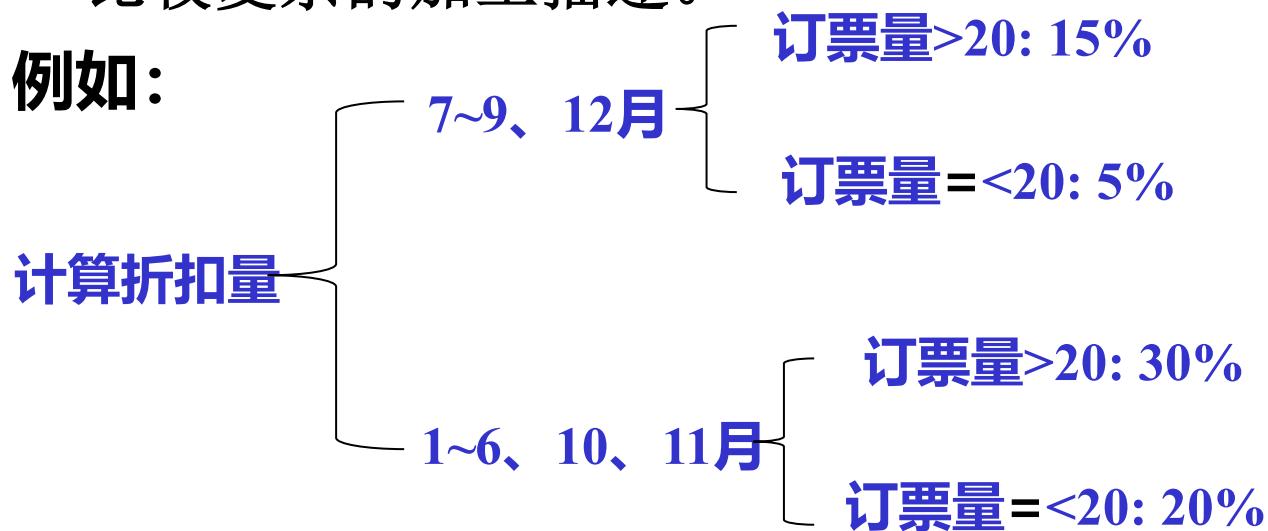
北京大学



③ 判定树:

适用于加工的输入数据和输出数据之间的逻辑关系比较复杂的加工描述。

例如:





5. 过程指导

① 建立系统的功能模型

---使用的工具为数据流图DFD

首先：建立**系统环境图(顶层数据流图)**，确定系统边界

继之：自顶向下，逐步求精，建立系统的层次数据流图

② 建立数据字典

---使用的工具为结构符：+、|、{ }等

定义数据流 定义数据存储

定义数据项

③ 给出加工小说明：**集中描述一个加工“做什么”，即加工逻辑，也包括其它一些与加工有关的信息，如执行条件、优先级、执行频率、出错处理等。**

---使用的工具可以为判定表 判定树



北京大学



结构化分析方法应用实例

问题 1：建立一个简化的商业自动化系统，其中：

营业员通过该系统记录每日销售的商品（商品名，商品编号，单价，数量，销售时间）；

收款员通过该系统记录收到的现金数额以及购物余额；

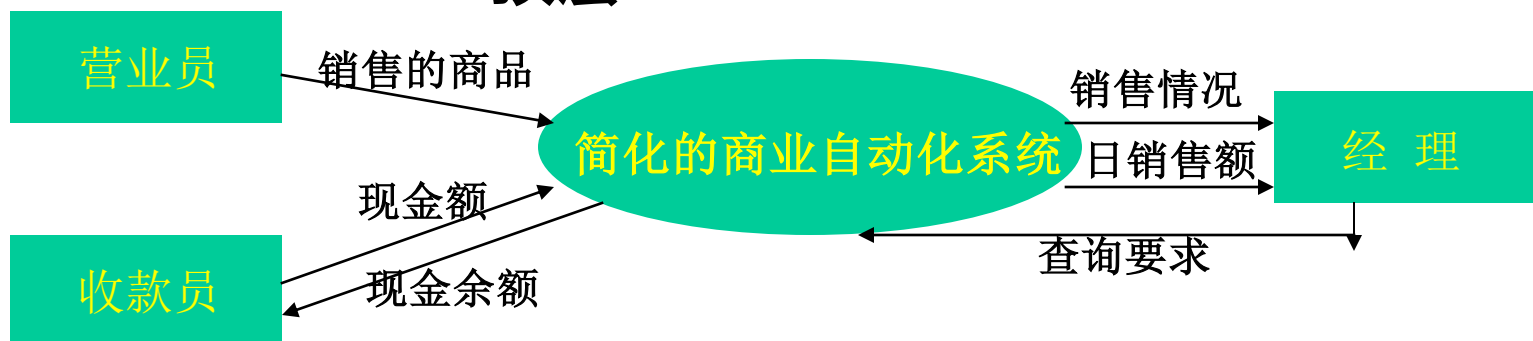
商店经理每日统计销售额，并在必要时查看某种商品的销售情况（商品名，商品编码，金额）。



①建立系统的功能模型

首先：建立系统环境图，确定系统边界

-----顶层DFD



其中：1 数据流为：销售的商品，日销售额等

3个输入流，3个输出流

数据源为：营业员，经理，收款员

数据潭为：经理，收款员

2 加工名为：要建立的系统名字



北京大学



继之：自顶向下，逐层分解

A、按人或部门的功能要求，将加工“打碎”（将“父图”的每一加工按其功能分解为若干子加工），形成：



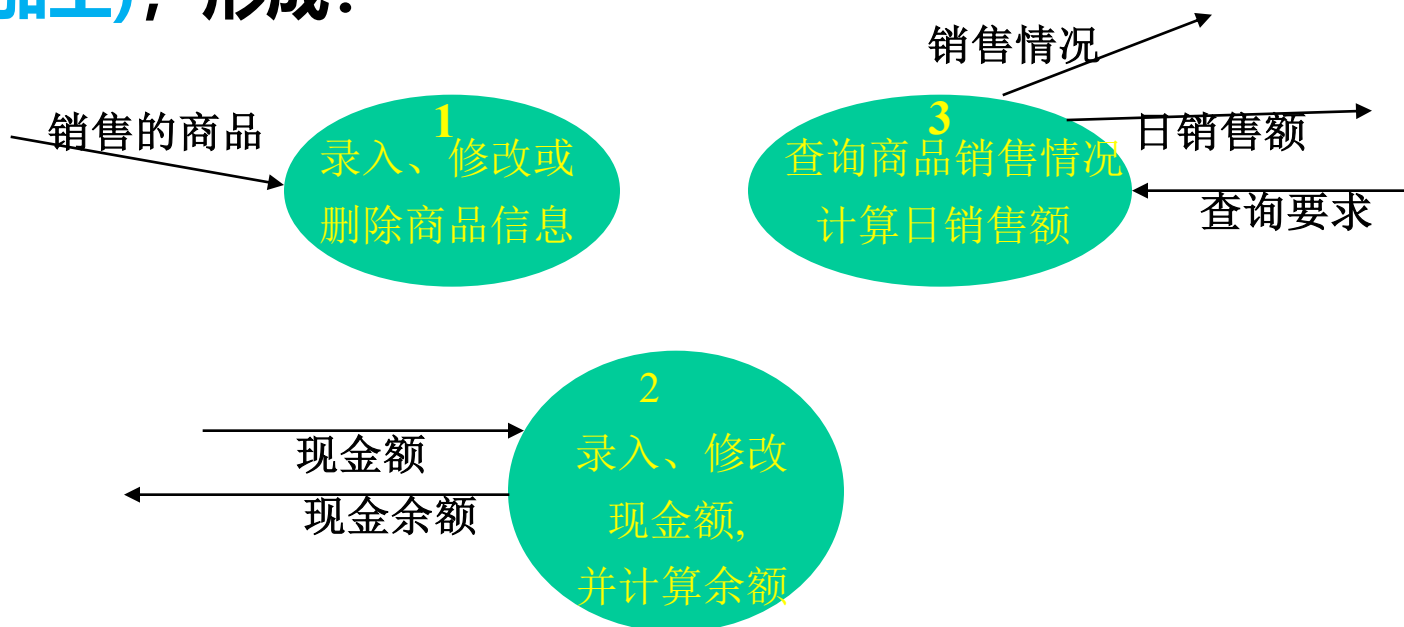
注：需给每一加工编号；



北京大学



B、“分派”数据流(将“父图”的输入流和输出流“分派”到子加工)，形成：



其中：要根据特定的加工要求进行分派；

保持与顶层数据流的一致；

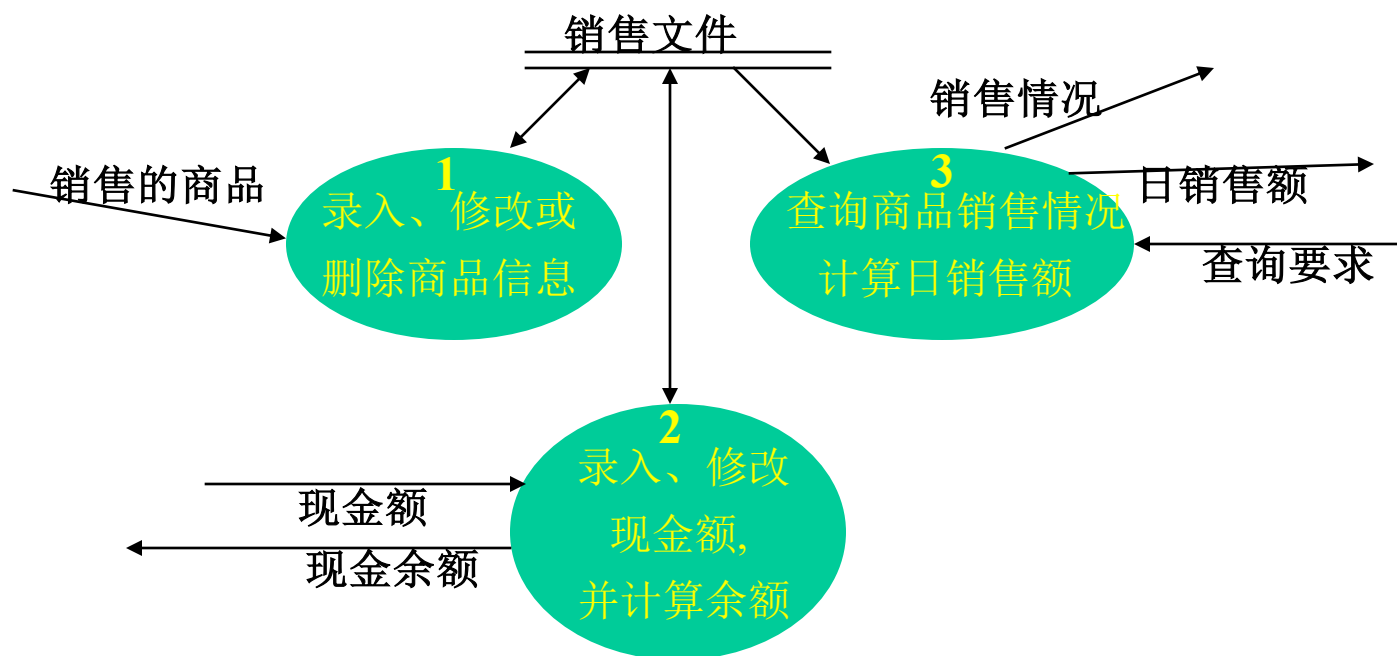
可以不引入数据源和数据潭。



北京大学



C、引入文件，使之形成一个有机整体—系统（在各加工之间建立合理的关系）：



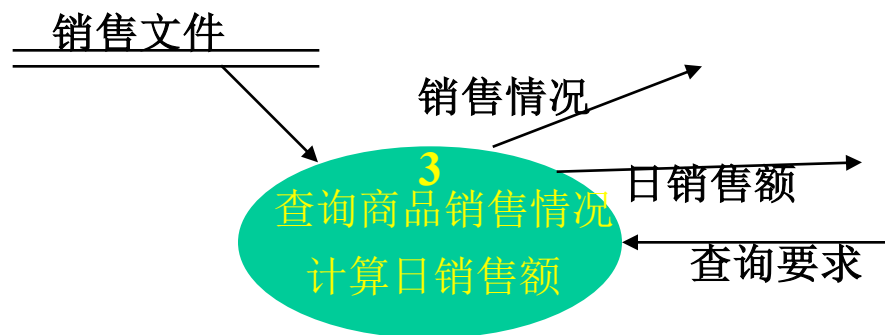
注：到一个文件，既有输入流，又有输出流，则可简化为
↔，并可不出标识。

至此，体现精化，形成0层数据流图。



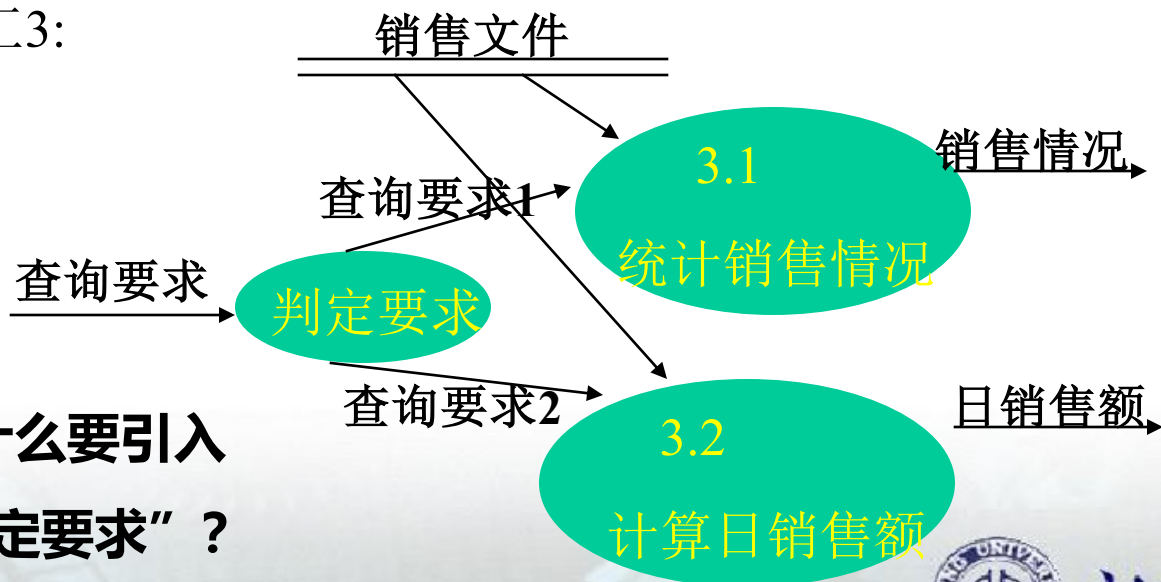
北京大学

继续A、B、C：自顶向下，逐层分解。例如：加工3



可分解为：

加工3:



*其中为什么要引入
加工“判定要求”？



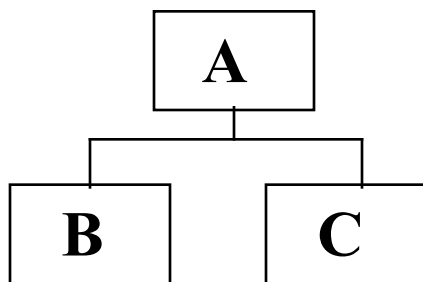
北京大学



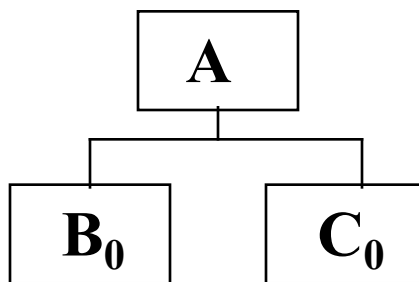
②建立数据字典

定义数据流 定义数据存储 定义数据项

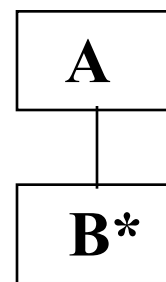
引入：结构符 + | {}
- 用于定义数据结构



(a) 顺序结构




(b) 选择结构



(c) 重复结构



北京大学



数据字典:

①、数据流:

销售的商品=商品名+商品编号+单价+数量+销售时间

现金额 = 余额 = 日销售额=非负实数

查询要求=[商品编号|日期]

查询要求1=商品编号

查询要求2=日期

销售情况=商品名+商品编号+金额

②、数据存贮:

销售文件={销售的商品}

③、数据项（数据流及数据存储的组成成分）

给出所有数据项的数据结构类型定义



北京大学



③给出加工小说明

描述一个加工，一般遵循如下模版：

加工编号：给出加工编号

加工名：给出该加工的标识


输入流：给出该加工的所有输入数据流

输出流：给出该加工的所有输出数据流

加工逻辑：采用结构化自然语言或判定表或判定树等工具，
给出该加工输入数据和输出数据之间的关系



北京大学



6. 建模中应注意的问题

结构化分析方法是一种半形式化的规约方法，给出了一组特定的术语表和标准化的表达格式-数据流图，在表达上均必须遵循一些约定，即应以一种准确和一致方式使用之。

例如：

--加工：其标识应使用领域术语，采用动宾结构；

必须有输入和输出；

--数据流：其标识应使用领域术语，采用名词或名词短语；

特别，应注意模型平衡问题和信息组织的复杂性控制问题，

即：



北京大学



①模型平衡问题

- **父图和子图的边界一致问题**，即它们应有相同的边界；
- **数据流图中的数据流与数据字典的一致问题**，即每个数据流和数据存储必须在数据字典中予以定义
- **数据流图中的“叶”加工（最低层）与小说明的一致问题**，即必须给出“叶”加工的说明
- **小说明和数据流图的图形表示一致问题**，即在小说明中，必须说明“输入数据流”如何使用，必须说明如何产生“输出数据流”；必须说明如何选取、使用、修改“数据存储”





②信息组织复杂性控制问题

- 上层数据可以“打包”

- 上、下层的数据流对应关系在数据字典中给出，但包内数据流的性质（输入、输出）必须一致。

- 一幅图中的图元个数应控制在 7 ± 2 以内

- 与每一加工相关的数据流的数目应适中

（与层次有关）

- 分析数据内容，确定是否所有的输入信息都用于产生输出信息；

- 分析加工，确定一个加工所产生的输出，是否都能由该加工的输入信息导出。





结构化分析方法应用实例

问题 2 :图书管理系统

图书管理系统旨在用计算机对图书进行管理，主要涉及五个方面的工作：新书入库、读者借书、读者还书以及图书注销，以及查询某位读者的借书情况、某种图书和整个图书的库存情况。

① 在购入新书时，图书管理人员为购入的新书编制图书卡片，包括分类目录号、流水号(要保证每本书都有唯一的流水号，即使同类图书也是如此)、书名、作者、内容摘要、价格和购书日期等信息，并写入图书目录文件中；

② 读者借书时，首先填写借书单，包括姓名、校园卡号、欲借图书分类目录号等信息，然后管理人员将借书单输入系统，继之系统检查该读者号是否有效，若无效，则拒绝借书；否则进一步检查该读者所借图书是否超过最大限制数(此处我们假设每位读者同时只能借阅不超过五本书)，若已达到最大限制数(此处为五本)，则拒绝借书；否则读者可以借出该书，登记图书分类目录号、读者号和借阅日期等，写入到借书文件中；





- ③ 在读者还书时，读者填写还书单，由管理人员将其输入系统后，系统根据其中的学号，从借书文件中读出该读者的借阅记录，获取该书的还书日期，判定该图书是否逾期，以便按规定做出相应的罚款；
 - ④ 在对一些过时或无继续保留价值的图书进行注销时，管理人员从图书目录文件中删除相关的记录；
 - ⑤ 当图书馆领导提出查询要求时，系统应依据查询要求，分别给出相应的信息。
- 其中假定，“为购入的新书编制图书卡片”，“读者首先填写借书单”等功能，均由人工实现。





根据问题陈述，按功能可以把该系统分为两“大块”，即：
借还书等事务的处理，以及咨询事务处理。

---进行功能抽象。

（注：不同的功能抽象将导致不同的结果！但应该是等价的。）

于是，可以根据这一抽象，可以识别：

1) 顶层数据流：借还书等事务处理要求

咨询事务要求

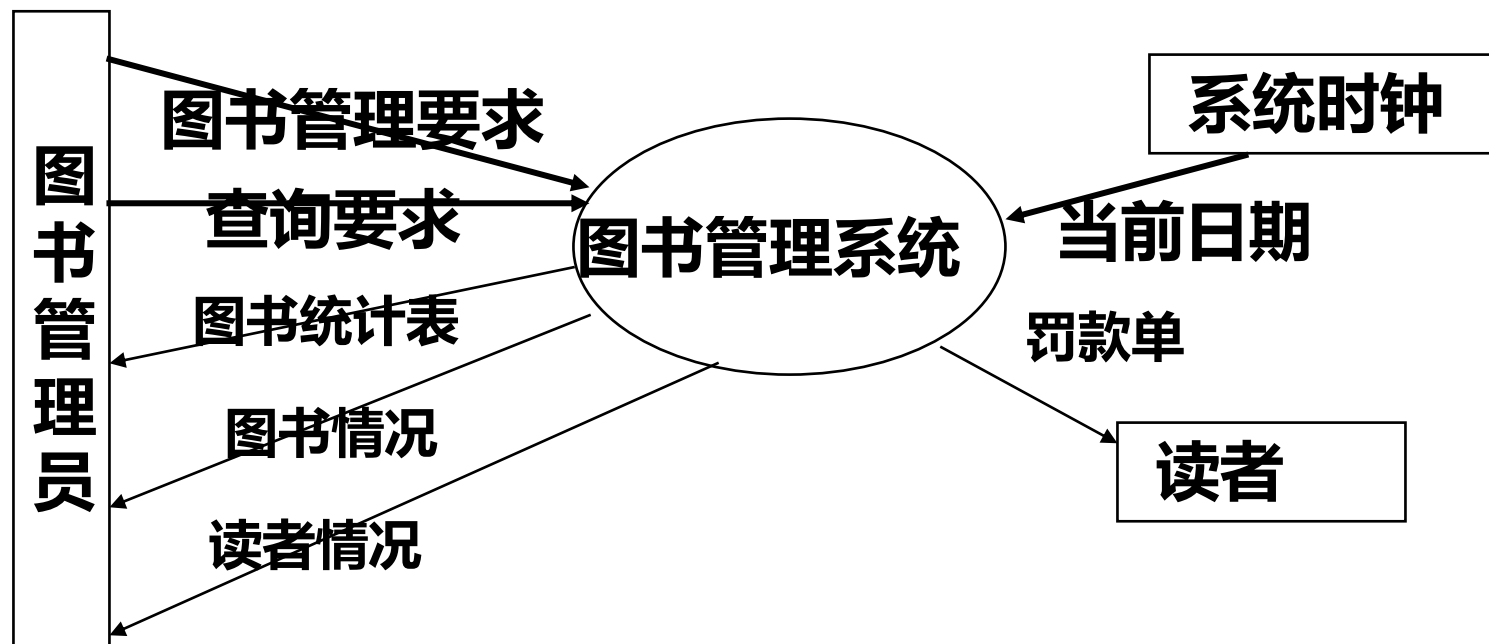
以及相关的数据流

2) 数据源和数据潭为：图书管理人员，读者以及时钟。



北京大学

(1) 建立系统的顶层数据流图（系统环境图）



其中：3个输入流：图书管理要求，查询要求，当前日期

图书管理要求=入库单 借书单 还书单 注销单

查询要求=读者情况 图书情况 图书统计表

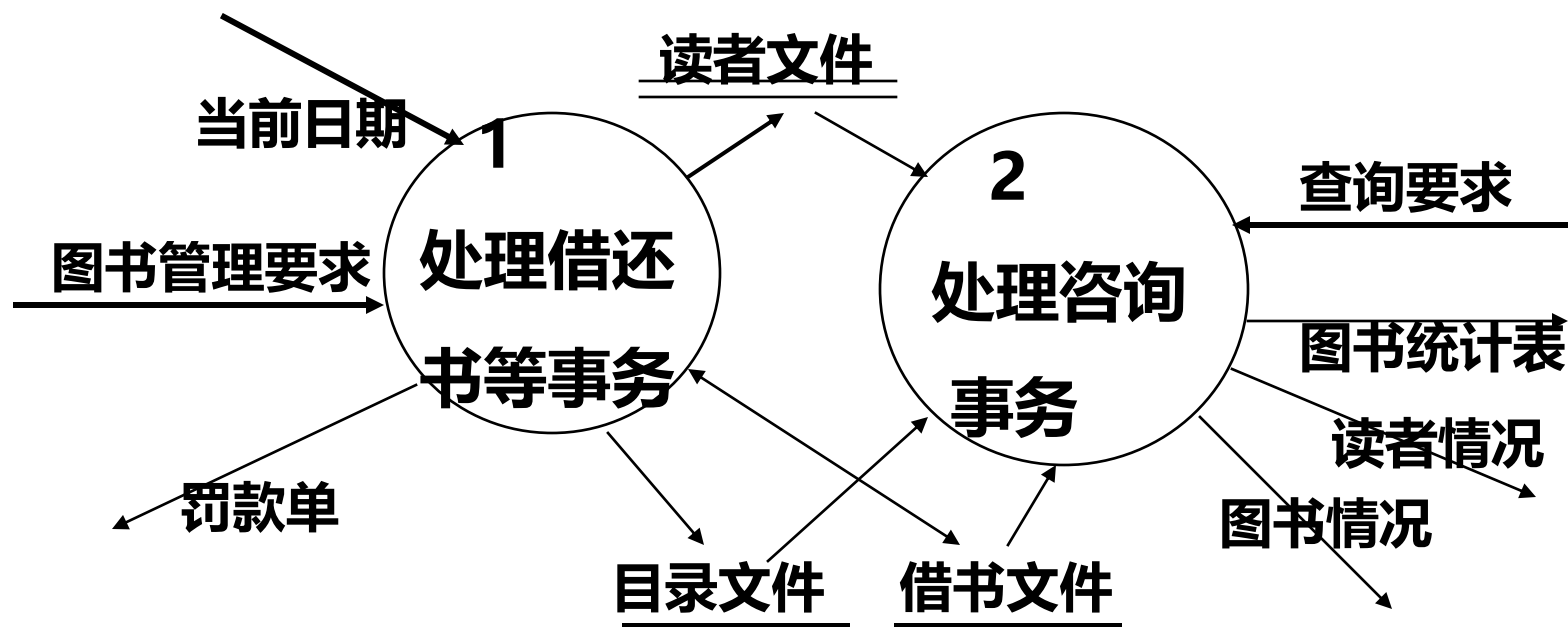
4个输出流：图书统计表，图书情况，读者情况，罚款单



北京大学

(2) 自顶向下，逐层分解

通过“打碎”、“分派”，可形成如下0层DFD：



其中：保持输入与输出的一致；

引入三个文件，对顶层DFD进行细化。

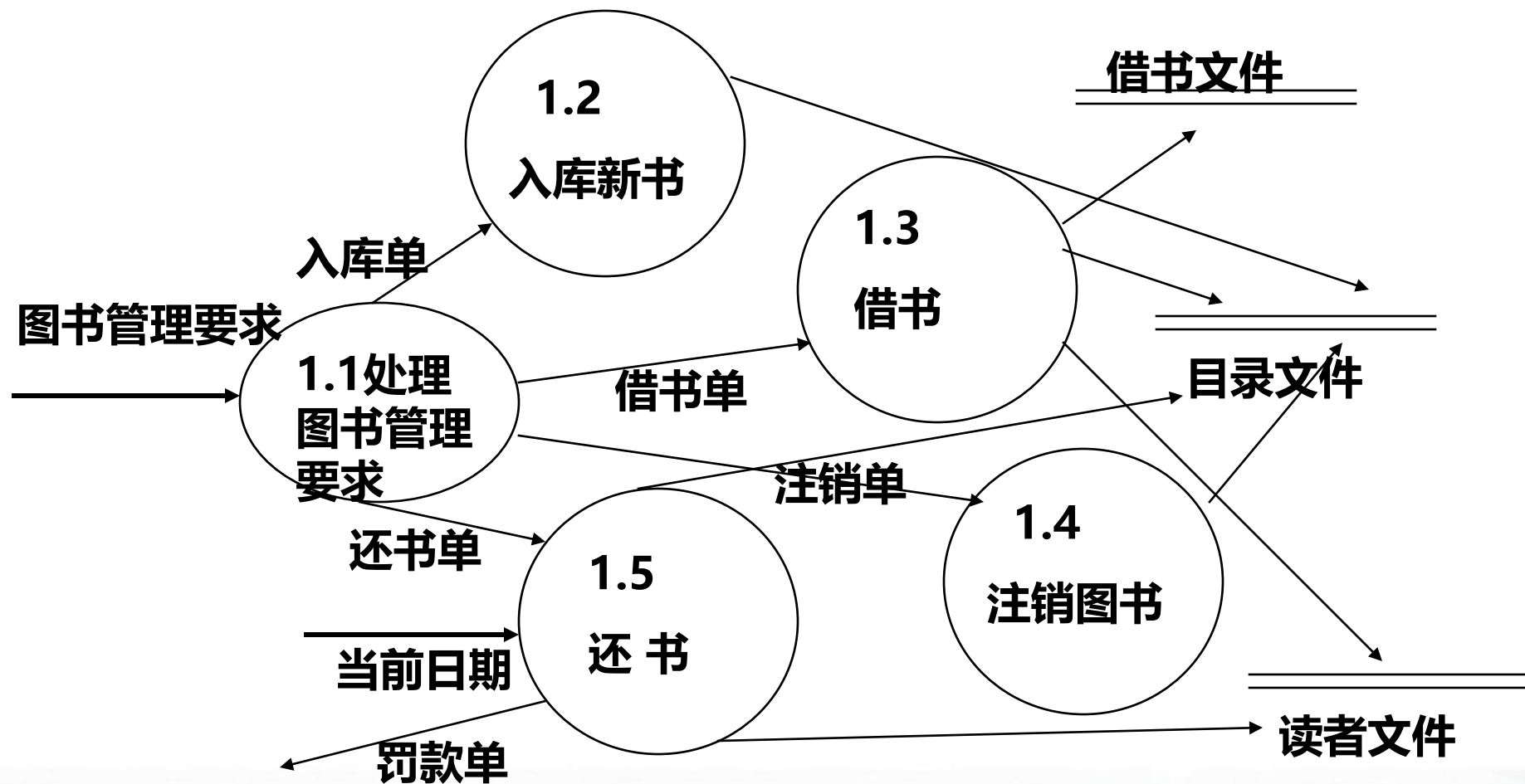
（注：存在数据库设计问题）



北京大学



以同样方式，对加工1进行分解，形成：



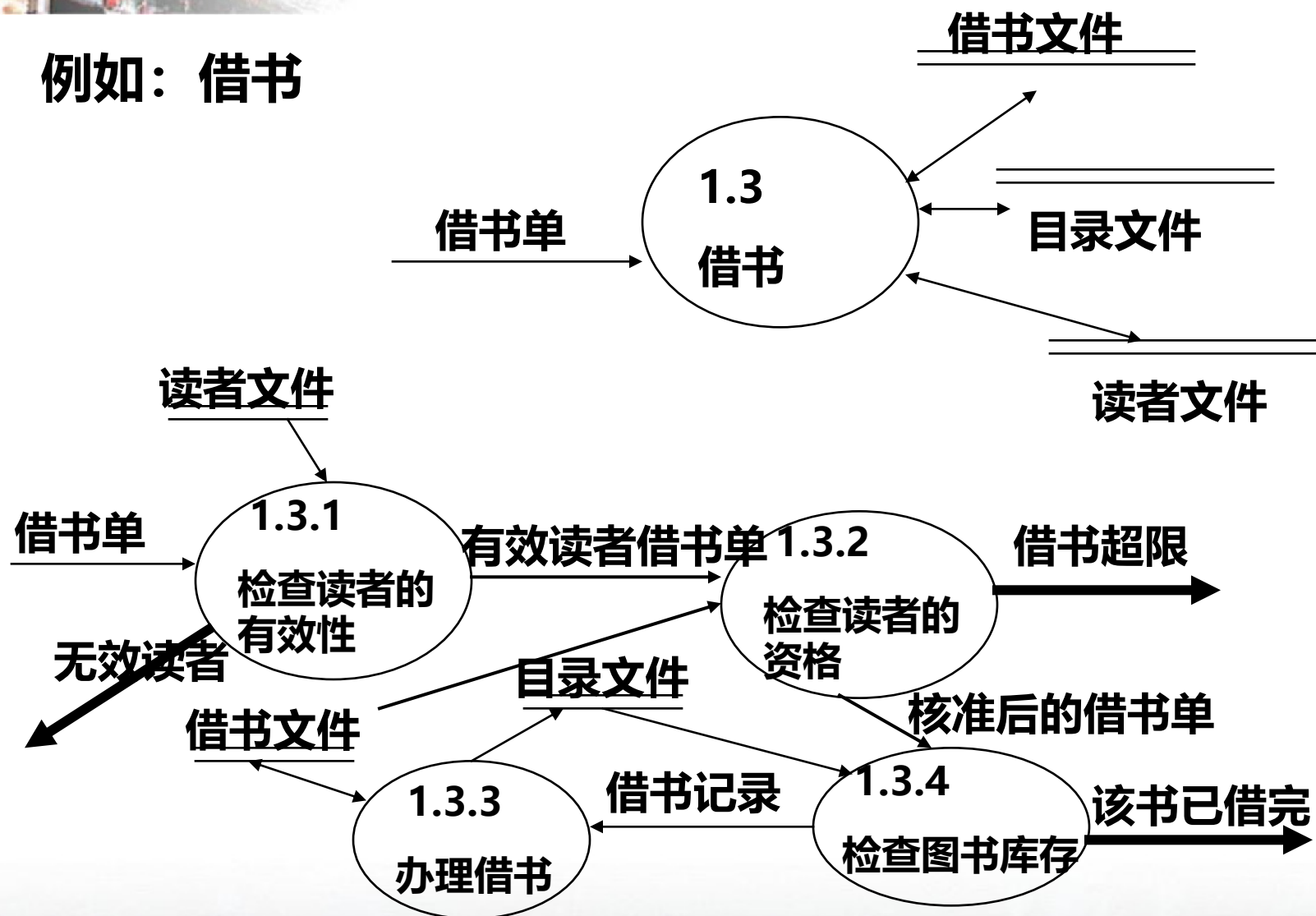
其中：注意平衡问题；平面化问题。当然，还可以继续细化



清华大学



例如：借书



注意：其中粗线数据流！



北京大学



(3) 建立系统的数据字典

数据流条目

查询要求=[读者情况|图书情况|图书统计表]

读者情况=姓名+校园卡号+可借图书数

图书情况=图书类号+图书流水号+书名+作者

图书统计表=[A|B|C|D|E]

图书管理要求=[入库单|借书单|还书单|注销单]

**入库单=图书类号+图书流水号+书名+作者+内容摘要+单价
+购书日期**

借书单=姓名+校园卡号+书名+图书类号+借书日期

还书单=姓名+校园卡号+书名+图书类号

注销单=图书类号+图书流水号+书名+价格+购书日期+单价

查询结果=读者情况|图书情况|图书统计表

.....



北京大学



数据存储条目：

借书文件={借书单}

目录文件={入库单}+库存量

读者文件={读者情况}

数据项条目：

{略}



北京大学



(4) 建立加工小说明

由于本例中的加工逻辑很简单，采用结构化自然语言描述加工即可。

如：加工2.1 查询读者借书情况，可描述如下：

加工标号：2.1

加工名：查询读者借书情况

输入流：读者校园卡号

输出流：读者借书情况

加工逻辑：

begin 根据读者校园卡号，在借书文件中获取该读者的借书记录；准备输出流中的数据，并输出之。

end.



北京大学



7. 需求分析的输出

XXX系统需求规格说明书

1. 引言

1.1 编写目的

说明编写本需求规格说明书的目的

1.2 背景说明

(1) 给出待开发的软件产品的名称;

(2) 说明本项目的提出者、开发者及用户;

(3) 说明该软件产品将做什么, 如有必要, 说明不做什么。

1.3 术语定义

1.4 参考文献



北京大学



2. 概述

2.1 功能概述

叙述待开发软件产品将完成的主要功能。

2.2 约束

叙述对系统设计产生影响的限制条件，并对下一节中所述的某些特殊需求提供理由，如管理模式、硬件限制、安全等。

3. 数据流图与数据字典及加工说明

3.1 数据流图

3.1.1 数据流图1

(1) 画出该数据流图

(2) 加工说明

(3) 数据流说明

3.1.2 数据流图2



北京大学



3.2 数据字典

3.2.1 文件说明

3.2.2 数据项说明

4. 接口

4.1 用户接口

4.2 硬件接口

4.3 软件接口

5. 性能需求

5.1 精度

逐项说明对各项输入数据和输出数据达到的精度

5.2 时间特征

定量说明本软件的时间特征，如响应时间、更新处理时间、数据传输、转换时间、计算时间等。



北京大学



5.3 灵活性

说明本软件所具有的灵活性，即当用户需求有某些变化时（如操作方式、运行环境、时间特征等），本软件的适应能力。

6. 属性

6.1 可使用性

规定某些需求，如检验点、恢复方式和重启动性，以确保软件可使用。

6.2 保密性

规定保护软件的要素

6.3 可维护性

6.4 可移植性





7. 其他需求

7.1 数据库

7.2 操作

7.3 故障及处理





8. 需求验证

有关SRS（**Software Requirements Specification**）内容方面：

- （1）**正确性**：指的是SRS中陈述的每个需求都表达了将要构造的系统的某个要求。
- （2）**无二义性**：指的是SRS中陈述的每个需求都只有一种解释。
- （3）**完整性**：若一个SRS具有以下三个特性，则它是完整的：
 - 未来系统所做的任何事情都包含在SRS的陈述中；
 - 未来系统响应所有可能的输入（包括有效和无效）；
 - SRS中没有被标识为“待定”的内容。





(4) 可验证性： SRS中陈述的每个需求都是可验证的—
即当且仅当存在一个有限代价的过程（人工或机器）
可以检查构造的软件产品是否符合用户的需求。

- 任何二义性必然导致不可验证性
- 任何不可度量的量，如“通常”或“时常”等
- 任何等同于停机问题的需求是不能被验证的，如“程序将不会进入一个无限循环”等同于停机问题，因而是不可验证的。

(5) 一致性：

- SRS中陈述的需求没有与以前的文档发生冲突；
- SRS中陈述的各个需求之间没有发生冲突。

(6) 可理解性：





有关SRS格式与风格方面：

(7) 可修改性：指的是SRS的结构和风格使任何对需求的必要修改都易于完整、一致的进行。

(8) 可被跟踪性：指的是SRS中的每个需求的出处都是清楚的，这意味着SRS中包含对前期支持文档的引用表。

(9) 可跟踪性：指的是SRS的书写方式有助于对其中陈述的每个需求进行引用。

(10) 设计无关性：指的是SRS不暗示特定的软件结构和算法。

(11) 注释：向开发机构提供了每个需求是否重要的指导意见。
如：E(Essential)、D(Desirable)或O(Optional)。



北京大学