

本章主要内容

- 软件需求的内容
- 需求分析的过程
- 需求分析的方法





需求分析的重要性



软件需求的定义和层次



需求工程



需求分析方法



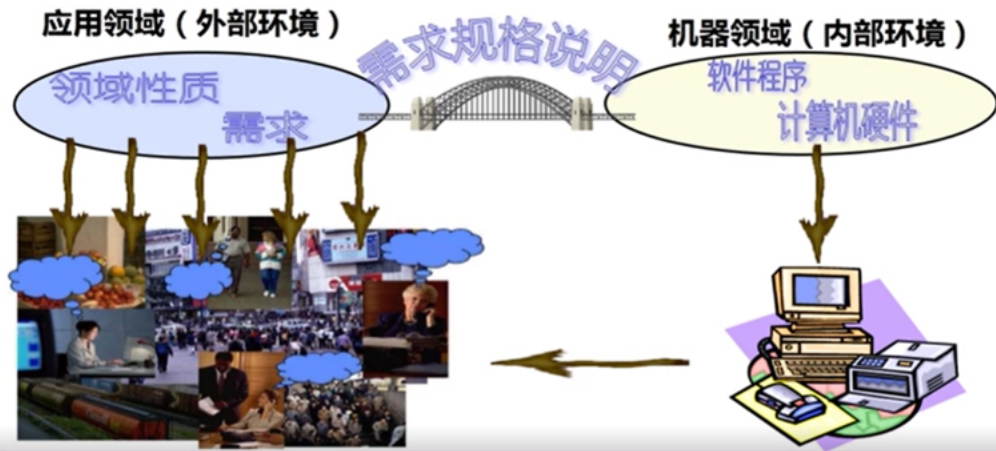
需求分析文档的原则和评审



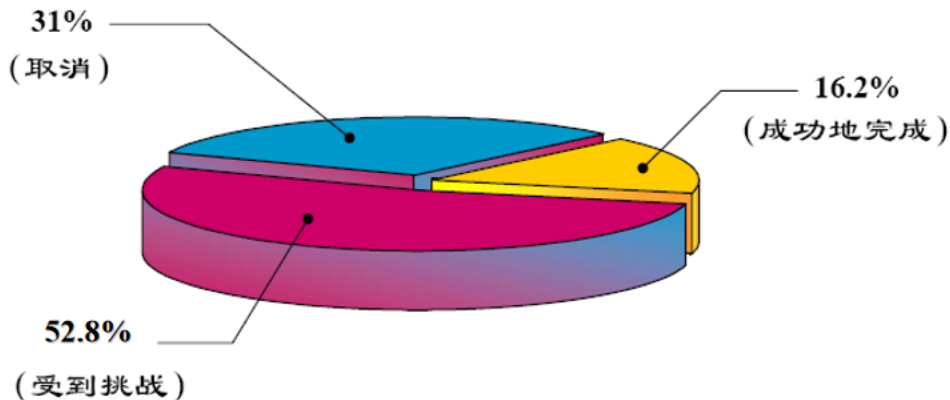
需求管理



需求，是人們要解決的某個問題或達到某種目的的需要。
是系統或其組成部門為滿足某種書面約定（合同、標準、規範等）所具有的能力。需求將作為系統開發、測試、驗收的正式文檔依據。 - IEEE 610.12, 1990

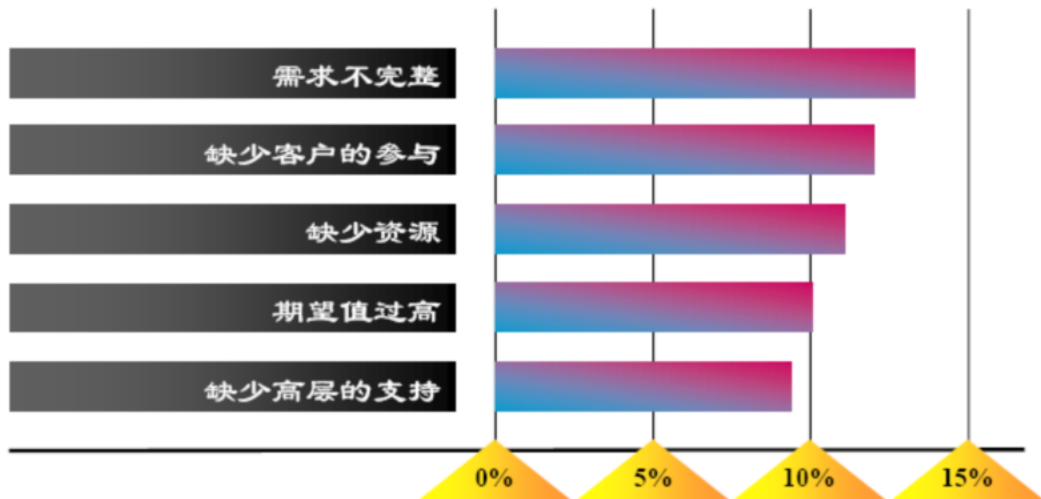


成功来之不易



软件项目失败的原因

软件项目失败的最重要的五个原因



一些常见的情景

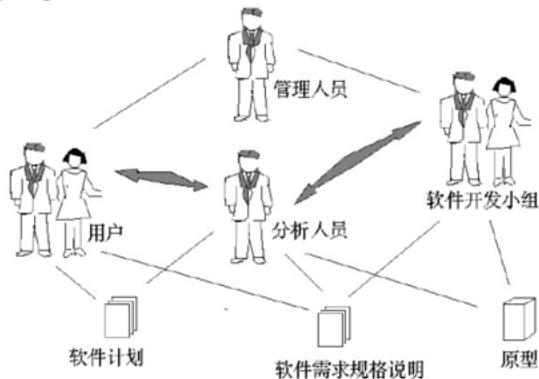
- 1) 开发人员在用户处呆了两三天就埋头开发;
- 2) 用户告诉开发人员我要开发一个XX系统, 但是我很忙, 你先开发一个让我看看;

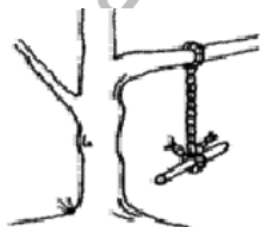
我知道你相信你已经理解了你认为我所说的内容, 但是我并不能肯定你已认识到你所听到的并不是我所想要的。



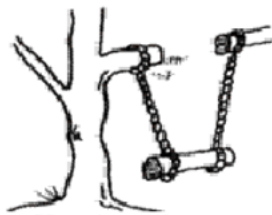
固有的困难

- 用户不一定清楚的知道他到底想要什么；
- 软件开发人员不了解项目的业务背景知识；
- 日常交流所用的语言文字本身具有很大的模糊性；
- 用户企业不同部门之间需求彼此矛盾；
- 用户的需求经常会发生变化。





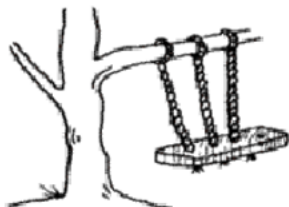
What the user asked for



How the analyst saw it



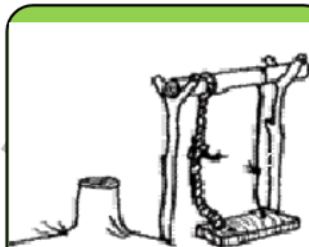
What the user really wanted



How the system was designed



As the programmer wrote it



How it actually works

软件需求是决定软件开发**是否成功**的一个关键因素：

- 需求分析可以帮助开发人员真正理解业务问题；
- 需求分析是估算成本和进度的基础；
- 需求分析可以避免建造错误的系统，从而减少不必要的浪费；
- 软件规格说明有助于开发人员与客户在“系统应该做什么”问题上达成正式契约；
- 软件需求是软件质量的基础，为系统验收测试提供了标准。





需求分析的重要性



软件需求的定义和层次



需求工程



需求分析方法



需求分析文档的原则和评审



需求管理



软件需求

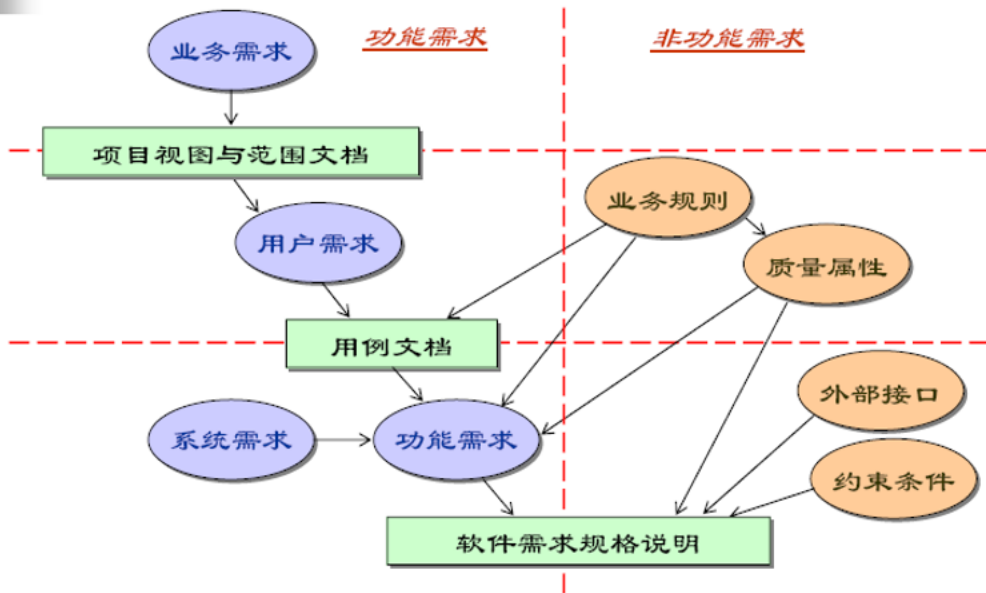
描述要“**做什么**”而不是“**怎样实现**”

需求分析阶段的研究对象是软件产品的用户需求。
这些需求最终要在所开发的软件产品上体现出来。

软件需求的概念涵盖了用户角度(系统的外部行为)和开发人员角度(系统的内部特性)两个方面，其中的关键在于需求一定要文档化。



软件需求的不同层次



业务需求

业务需求是组织或客户对于系统的高层次目标要求，定义了项目的远景和范围，即确定软件产品的发展方向、功能范围、目标客户和价值来源。

业务需求的内容

- 业务：产品属于哪类业务范畴？应该完成什么功能？
- 客户：产品为谁服务？目标客户是谁？
- 特性：产品区别于其他竞争产品的特性是什么？
- 价值：产品的价值体现在什么方面？
- 优先级：产品功能特性的优先级次序是什么？



业务需求：例

业务要求

- 各种图书资料的借阅、查询和管理；
- 使用计算机实现图书资料的日常管理，提高工作效率和服务质量；
- 用户通过网络查询和浏览电子资料，改变原有的借阅模式；
- 由于版权的限制，某些电子资料只能让用户浏览和打印而不能下载。

客户与用户

- 学院的高层管理者
- 图书管理员
- 借阅者：教师、学生



用户需求

用户需求是从用户角度描述的系统功能需求和非功能需求，通常只涉及系统的外部行为，而不涉及系统的内部特性。

用户需求的描述

- 原则：应该易于用户的理解。一般不采用技术性很强的语言，而是采用自然语言和直观图形相结合的方式描述。



系统需求

系统需求是更加详细地描述系统应该做什么，通常包括许多不同的分析模型，诸如对象模型、数据模型、状态模型等。

系统需求主要是面向开发人员进行描述，是开发人员进行软件设计的基础。

功能需求

功能需求：描述系统应该提供的功能或服务，通常涉及用户或外部系统与该系统之间的交互，一般不考虑系统的实现细节。



非功能需求

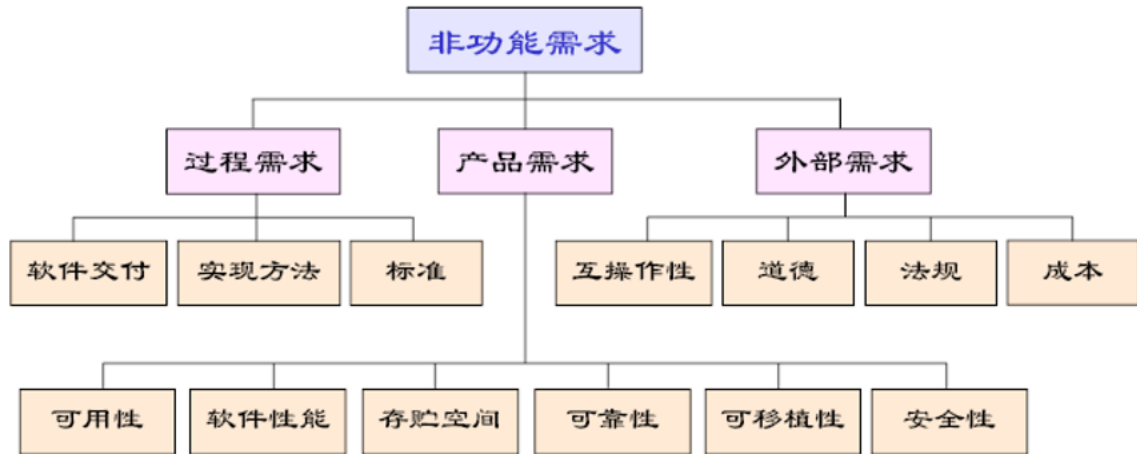
非功能需求：从各个角度对系统的约束和限制，反映了应用对软件系统质量和特性的额外要求，例如响应时间、数据精度、可靠性、开发过程的标准等。

举例：

- 系统应在20 秒之内响应所有的请求。
- 系统每周7 天、每天24 小时都可以使用。
- 对于一个没有经验的用户而言，经过两个小时的培训就可以使用系统的所有功能。



非功能需求



非功能需求

特性	度量指标
速度	- 每秒处理的事务 - 用户或事件的响应时间 - 屏幕的刷新时间
存储空间	- 字节数 - RAM 芯片数
易用性	- 培训时间 - 帮助页面数
可靠性	- 平均失败时间 - 系统无效的概率 - 失败发生率
容错性	- 失败后的重启次数 - 事件引起失败的比例 - 失败时数据崩溃的可能性





需求分析的重要性



软件需求的定义和层次



需求工程



需求分析方法



需求分析文档的原则和评审



需求管理

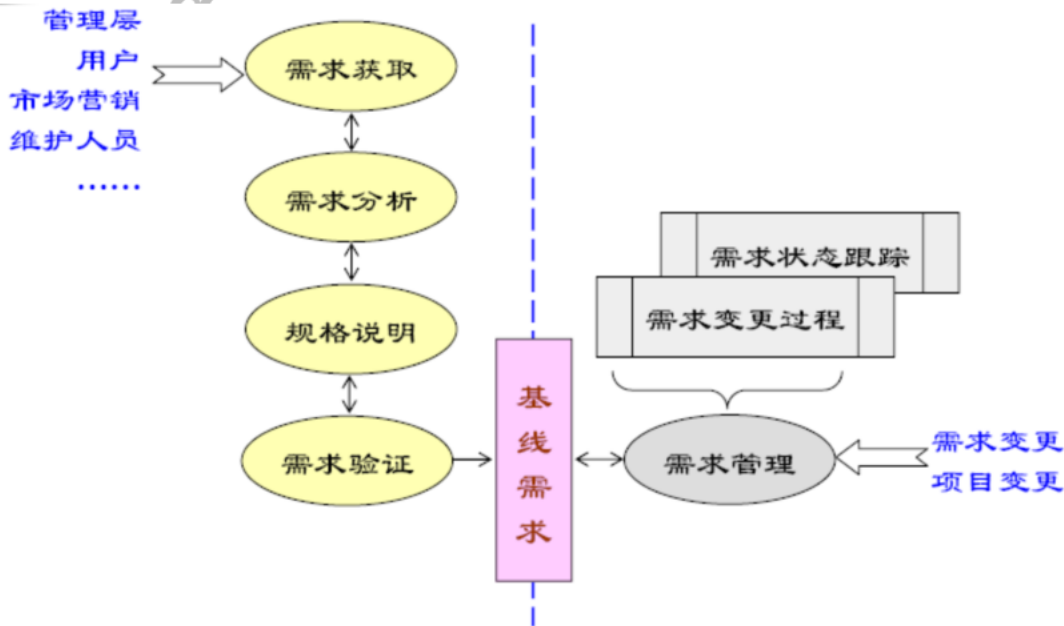


需求工程

需求工程是应用已证实有效的原理和方法，并通过合适的工具和符号，系统地描述出待开发系统及其行为特征和相关约束。

- 需求获取：开发人员聆听客户的需求，观察用户的行为；
- 需求分析：分析和整理所收集的用户需求；
- 需求规格说明：以文档形式，精确地阐述一个软件系统必须提供的功能和性能以及它所要考虑的限制条件；
- 需求验证：使用评审和商议等有效手段对其进行验证，最终形成一个需求基线；
- 需求管理：在软件开发过程中有效地管理和控制需求变更。





需求获取

- 需求获取是在问题及其最终解决方案之间架设桥梁的第一步。
- 需求获取的目的是清楚地理解所要解决的问题，完整地获得用户的需求。
- 获取需求是对项目中描述的客户需求的普遍理解。一旦理解了需求，分析者、开发者和客户就能探索出描述这些需求的多种解决方案。

eadonly



需求获取

- 需求获取包括以下活动：
 - 1) 发现和分析问题** 发现问题症结，并分析问题的原因/结果关系。
 - 2) 获取需求** 根据对问题的理解定义需求。
 - a) 使用调查研究方法收集信息；**
 - b) 遵循需求获取框架，按照三个成分观察：即数据、过程和接口。**
 - 3) 需求归档** 以草稿形式归档调查结果。形式有用例、决策表、需求表等。



需求获取技术

- 需求获取的关键在于通过与用户的沟通和交流，收集和理解用户的各项要求。
- 需求获取技术
 - 用户面谈
 - 需求专题讨论会
 - 问卷调查
 - 现场考察
 - 原型化方法
 - 基于用例的方法



需求获取注意点

CSDN: 你对需求分析怎么看? 你后来还提出了“客户需求可行性分析”理论, 能大概介绍其关键点吗?

<https://fangang.iteye.com/category/200994>

范钢: 这些年做了那么多软件项目, 有成功的、有失败的, 真的切身感受到, 需求分析的成功与否, 非常关键地决定了一个项目的成功与否。无数令人绝望的项目都是始于不到位的需求分析。然而, 在实际工作中, 我们的需求分析工作既低效又无所适从。通过这一系列的文章, 我点出了需求分析工作问题的核心, 那就是被动式的需求分析, 即客户说什么就是什么, 一切都以客户说的为准。客户往往是非技术的, 因此客户提出的需求往往也是非常理想而难以实现的。用这样的需求去指导开发, 结果可想而知。

正确的做法应当是主动式的需求分析, 即跳出客户需求的本身去主动学习与理解客户的业务领域, 分析客户需求背后的动机, 从而站在更加深刻的角度去理解客户的需求, 然后结合技术实现去分析客户需求隐藏的问题, 这就是所谓的“客户需求可行性分析”, 即分析客户需求是否可行。最后, 站在客户的角度去探讨其中的问题, 同时提出更加可行的方案。

这里的关键问题就在于我们对客户业务领域的理解到底有多深。我们对业务领域理解得越深, 我们对客户需求的理解就越深, 分析的问题也就越深, 与客户探讨的过程中就越能被客户认可, 从而提出为客户认可与接收的技术方案。这样的方案, 既准确表达了客户的需求, 又可以在技术上是可行的。按照这样的方案设计开发软件项目, 才能真正有效地规避需求方面的风险。这种思路在实践中得到了许多的成功, 并且让无数的网友脑洞大开。



需求分析-建模

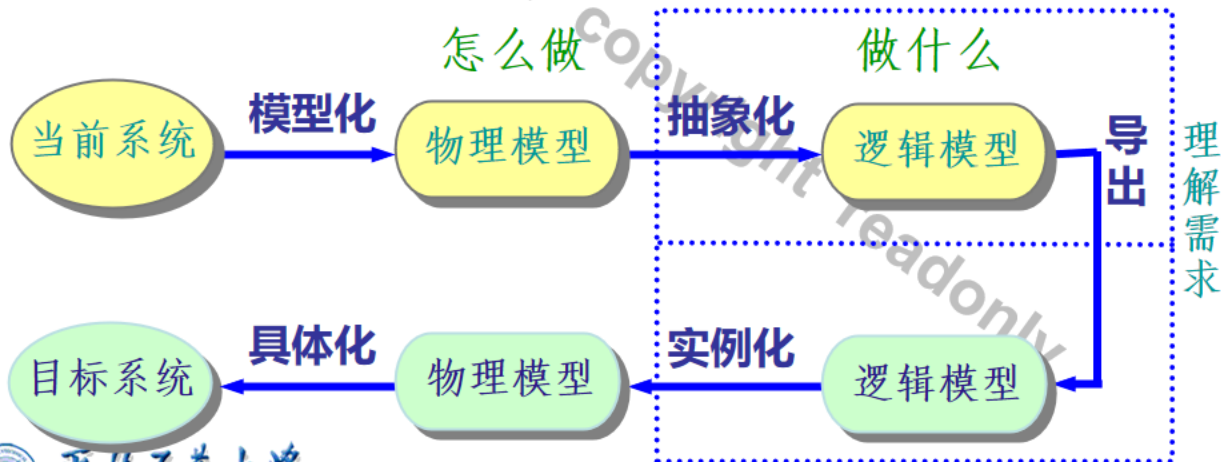
- 需求建模是为了分析需求，以确定项目的确切需求。
- 需求建模遵循三个原则：
 - 划分：描述需求的**整体——部分**关系；
 - 抽象：描述需求的**一般化——特殊化**关系；
 - 投影：描述需求的多维视图；
- 定义系统模型要区分逻辑模型和物理模型。
- 常用模型有数据建模和过程建模。

only



需求分析-建模

软件开发是要实现目标系统的物理模型。需求分析的任务就是借助于当前系统的物理模型导出目标系统的逻辑模型，解决目标系统“**做什么**”的问题。



需求分析-建模

	物理模型 (实施模型、 技术模型)	逻辑模型 (本质模型、 概念模型)
当前 系统	描述现实系统是如何在物理上实现的。	描述重要的业务功能，不考虑系统是如何实施的。
目标 系统	描述新系统是如何实施的（包括技术）。	描述新系统的主要业务功能和用户新的需求，不考虑系统应如何实施。

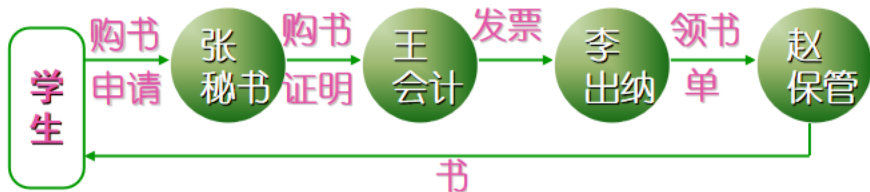


举

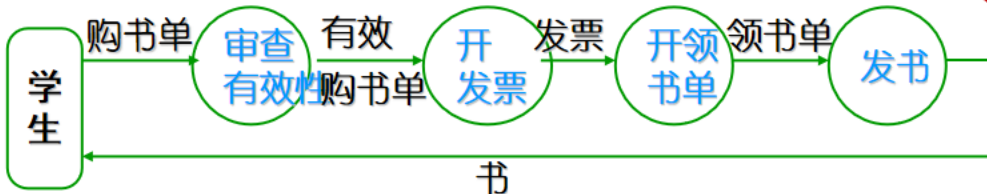
例

需求分析-建模

售书系统的逻辑模型



----- 学生购买教材的物理模型

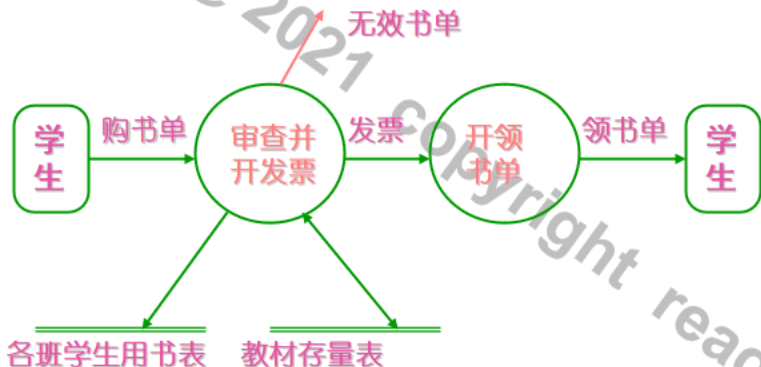


----- 学生购买教材的逻辑模型



需求分析-建模

售书系统的数据流图



需求分析-建模

需求建模的步骤

- 进行各种要求的一致性检查；
- 逐步细化所有的软件功能；
- 分解数据域，分配给各个子功能；
- 找出系统各成分之间的联系、接口特性和设计限制。
- 判断是否存在不合理的用户要求或用户尚未提出的潜在要求。
- 综合成系统的解决方案，给出目标系统的详细逻辑模型。



需求规格说明

- 又称需求定义或需求规约。
- 需求定义的主要目的是分析需求草稿和模型，解决其中存在的二义性和不一致性，系统地准确地表达系统需求，形成需求规格说明。包括
 - 系统应提供的功能和服务；
 - 非功能需求；
 - 系统开发或运行的限制条件；
 - 与系统互连的其他系统的信息。



需求验证

- 又称**需求评审**。需求验证的目的是确保需求编写正确。必须全面理解用户的各项要求，**准确表达**被接受的用户要求。
- 可能的错误有：
 - 不正确的系统模型；
 - 排版错误或语法错误；
 - 互相矛盾的需求；
 - 有二义性或用词不当的需求；
 - 没有遵循文档编制规范要求的质量标准。



需求管理

- 需求管理就是管理需求变化的过程。
- 需求管理涉及需求变更如何被处理的策略、规程和过程。它规定了
 - 应如何提交一个需求变更请求？
 - 如何分析需求变更对范围、进度和成本的影响？
 - 如何批准或驳回需求变更？
 - 如果批准了变更，该变更如何实现？

only





需求分析的重要性



软件需求的定义和层次



需求工程



需求分析方法



需求分析文档的原则和评审



需求管理



常用的分析方法

- 面向数据流的结构化分析方法 (SA)
- 面向对象的分析方法 (OOA) 等



结构化分析方法

- 结构化分析方法 (Structured Analysis, 简称SA方法) 是70年代中期提出的一种面向数据流、自顶向下、逐步求精进行需求分析的方法。
- 结构化分析方法适用于分析大型的数据处理系统, 特别适用于企事业管理系统。
- 结构化分析方法通常与设计阶段的结构化设计方法 (Structured Designed, 简称SD方法) 衔接起来使用。

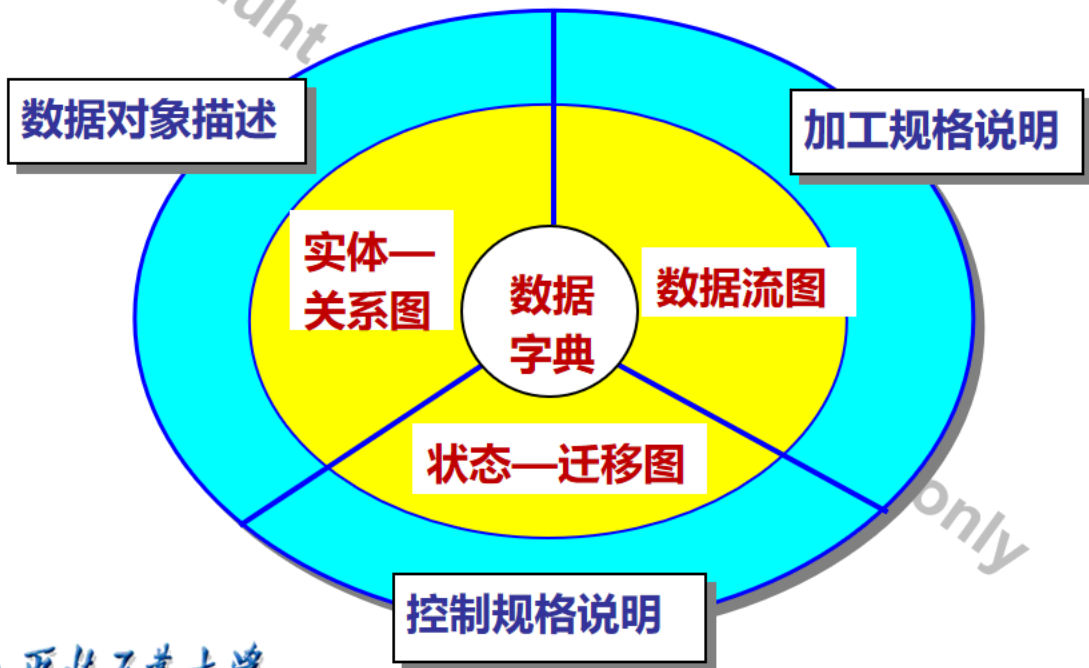


结构化分析方法

- 结构化分析方法最初只是着眼于数据流，自顶向下，逐层分解，建立系统的处理流程，以数据流图和数据字典为主要工具，建立系统的逻辑模型。
- 扩充后，将建模技术扩展到数据建模、功能建模和行为建模，以实体-关系图、数据流图和控制流图、状态-迁移图工具，数据字典为核心，从不同视点建立系统的分析模型。



结构化分析的分析模型



分析模型的主要描述工具

数据字典(DD)

- 模型核心，包含了所有数据对象的描述的中心库。

E-R图(ERD)

- 表示数据对象以及相互的关系，用于数据建模。

数据流图(DFD)

- 指明数据在系统中移动时如何被变换;
- 描述对数据流进行变换的功能;
- DFD中每个功能的描述包含在加工规约(小说明)。
- 用于功能建模。

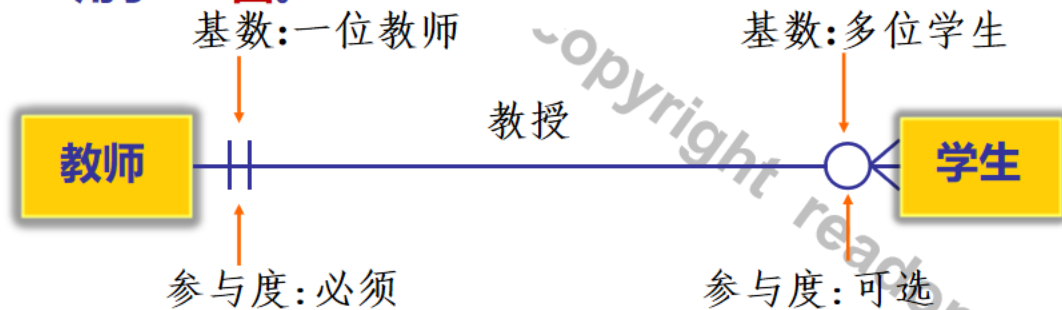
状态变迁图(STD)

- 指明作为外部事件的结果,系统将如何动作。用于行为建模。



数据对象建模--- E-R图

- 数据模型包括三种互相关联的信息：**数据对象**，描述对象的**属性**，描述对象间相互连接的**关系**。
- 在需求分析阶段描述数据对象和它们之间的关系，使用了**E-R图**。

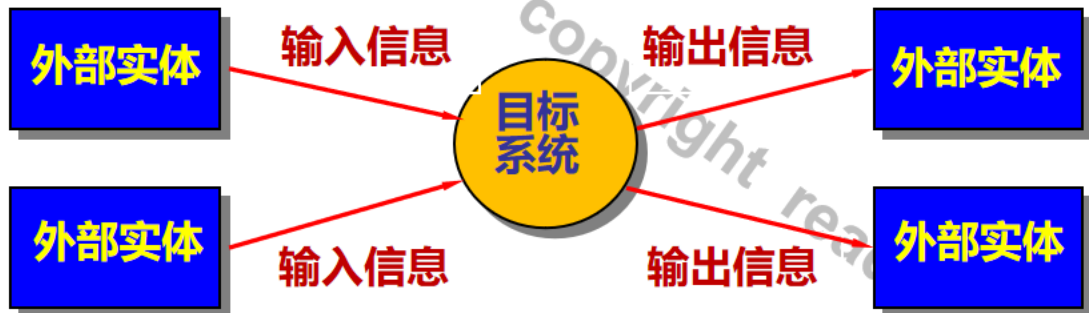


例如，在教学管理中，一个教师可以教授一门或多门课程，每位学生也需要学习几门课程。因此，教学管理中涉及的对象(实体型)有学生、教师和课程。



功能建模---数据流图

目标系统被表示成如图所示的数据变换流程图。
系统的功能体现在核心的数据变换中。



功能建模---数据流图

- 功能建模就是用抽象模型的概念，按照软件内部**数据传递、变换**的关系，**自顶向下**逐层分解，直到找到满足功能要求的所有可实现的软件为止。
- 功能模型使用了**数据流图**来表达系统内数据的运动情况，而数据流的变换则用结构化语言、判定表与判定树来描述。



功能建模---数据流图

数据流图(Data Flow Diagram, DFD)是描述系统中数据流程的图形工具, 它标识了一个系统的逻辑输入和逻辑输出, 以及把逻辑输入转换为逻辑输出所需的加工处理。

数据流图的图符, 四种基本图形符号:

一些辅助的图例:

数据流

数据流名 箭头

加工

加工名 圆或椭圆

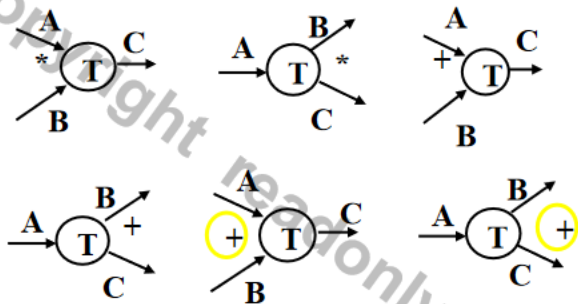
数据存储

单或双杠
文件名

数据源点
或终点

实体名

矩形框



* 与

+ 或

(+) 互斥

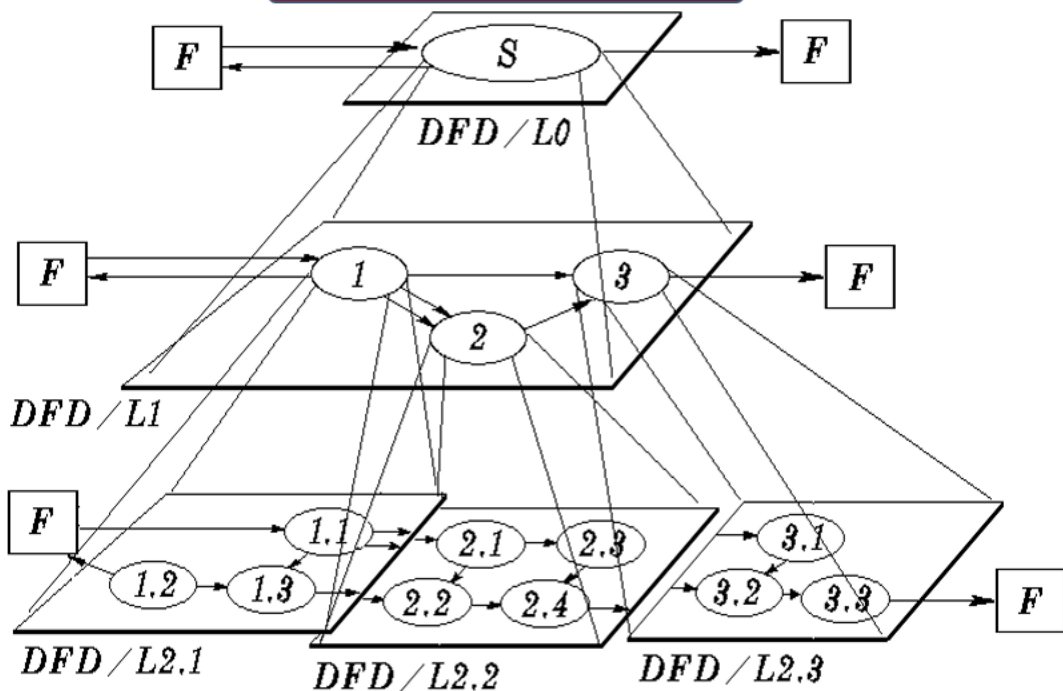


数据流图的层

- 为了表达数据处理过程，需要采用**层次结构**的数据流图。按照系统的**层次进行逐步分解**，并以分层的数据流图反映这种结构关系。
- **顶层**数据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**数据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**数据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。



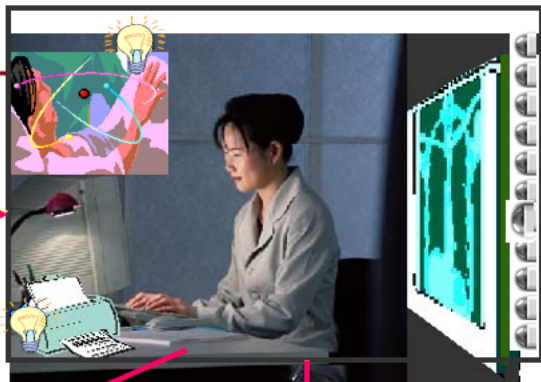
分层的数据流图



需求分析

实例：医院病房监护系统

监视病情



产生病情报告



经过初步的需求分析，得到系统功能要求：

- 1、监视病员的病症(血压、体温、脉搏等)。
- 2、定时更新病历。
- 3、病员出现异常情况时报警。
- 4、随机地产生某一病员的病情报告。

更新病历

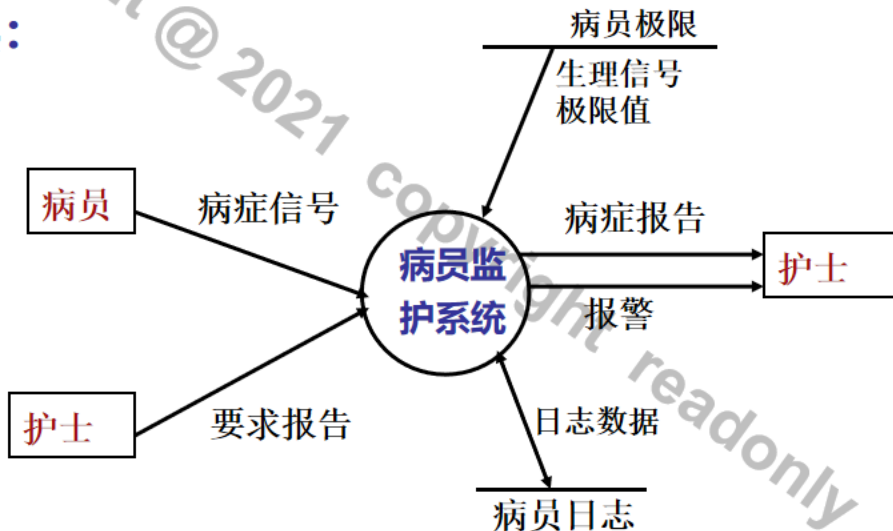


西北工业大学

NORTH WESTERN POLYTECHNICAL UNIVERSITY

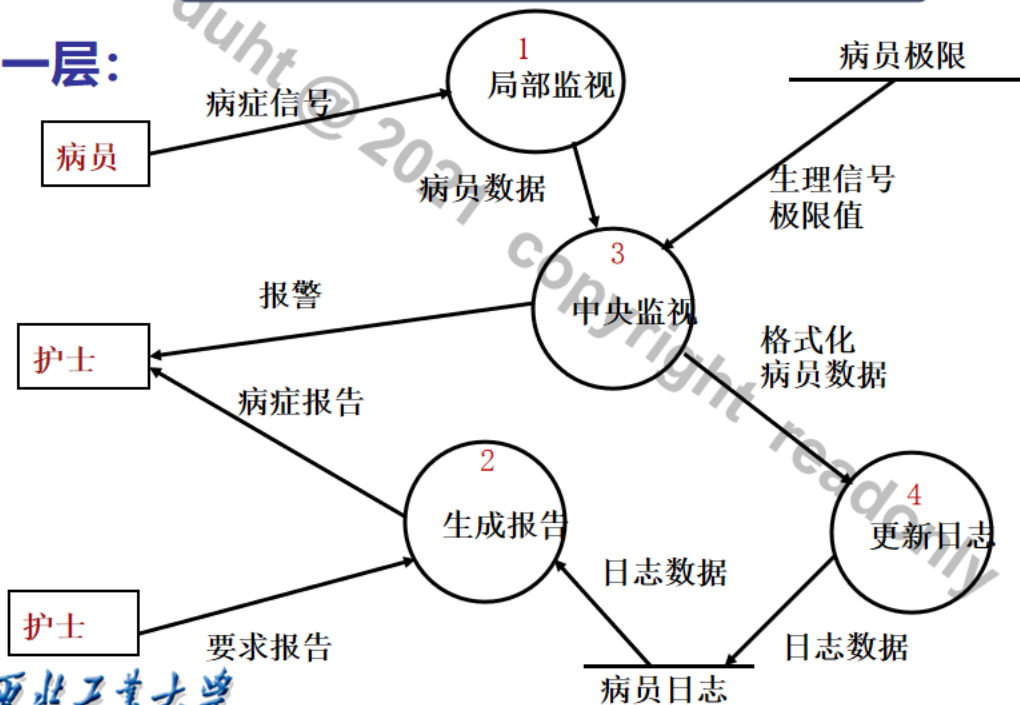
医院病房监护系统---数据流图

顶层:



医院病房监护系统---数据流图

第一层:



第二层：加工“中央监视”分解

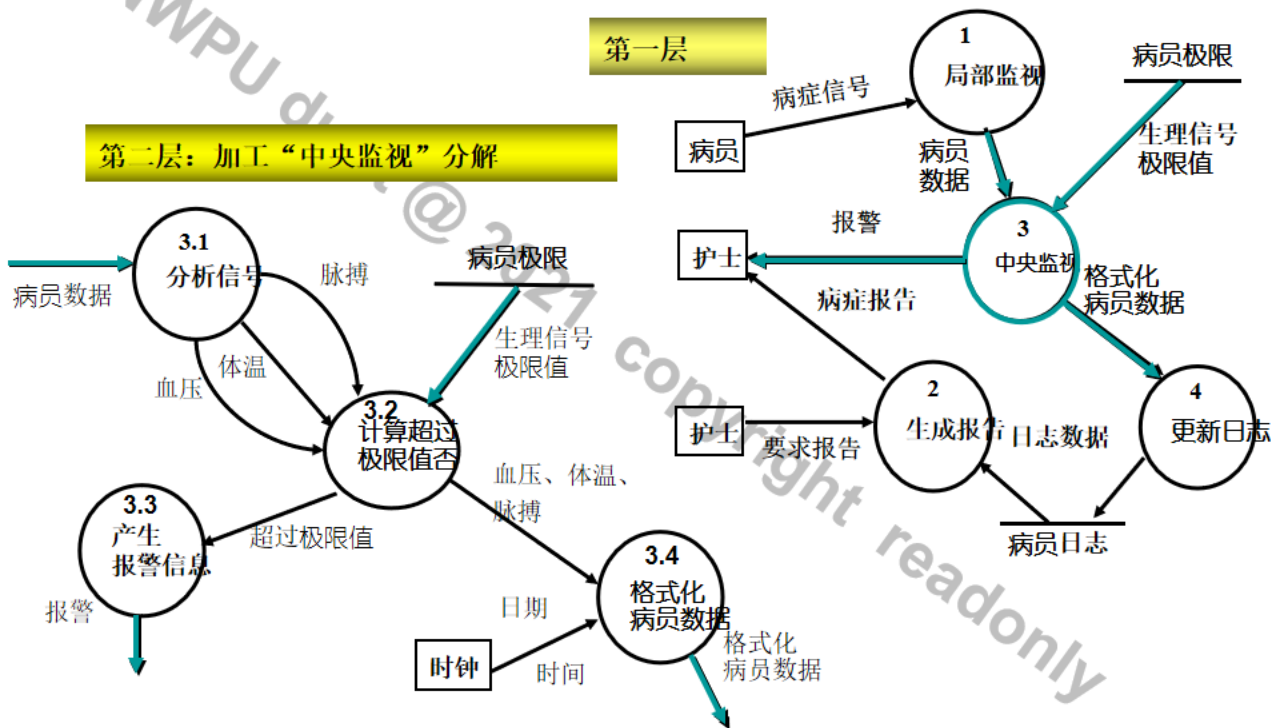


需求分析

需求分析方法

第一层

第二层：加工“中央监视”分解



画分层DFD图的基本原则

- **数据守恒与数据封闭原则**

所谓数据守恒是指加工的输入输出数据流是否匹配，即每一个加工既有输入数据流又有输出数据流。或者说一个加工至少有一个输入数据流，一个输出数据流。

数据封闭是对整个系统而言。

- **加工分解的原则**

自然性：概念上合理、清晰；

均匀性：理想的分解是将一个问题分解成大小均匀的几个部分；

分解度：一般每一个加工每次分解最多不要超过7个子加工，分解应分解到基本加工为止。



画分层DFD图的基本原则

- 子图与父图的“平衡”

父图中某个加工的输入输出数据流应该同相应的子图的输入输出相同(相对应)，分层数据流图的这种特点称为子图与父图“平衡”。

- 合理使用文件

当文件作为某些加工之间的交界面时，文件必须画出来，一旦文件作为数据流图中的一个独立成份画出来了，那么他同其他成份之间的联系也应同时表达出来。



DFD图不是流程图，不表示软件的控制流程。



分层DFD图的改进

DFD图必须经过**反复修改**，才能获得最终的目标系统的逻辑模型(目标系统的DFD图)。可从以下方面考虑DFD图的改进：

1、检查数据流的正确性

- ① 数据守恒
- ② 子图、父图的平衡
- ③ 文件使用是否合理。特别注意输入/出文件的数据流。

2、改进DFD图的易理解性

- ① 简化加工之间的联系（加工间的数据流越少，独立性越强，易理解性越好）。
- ② 改进分解的均匀性。
- ③ 适当命名（各成分名称无二义性，准确、具体）。

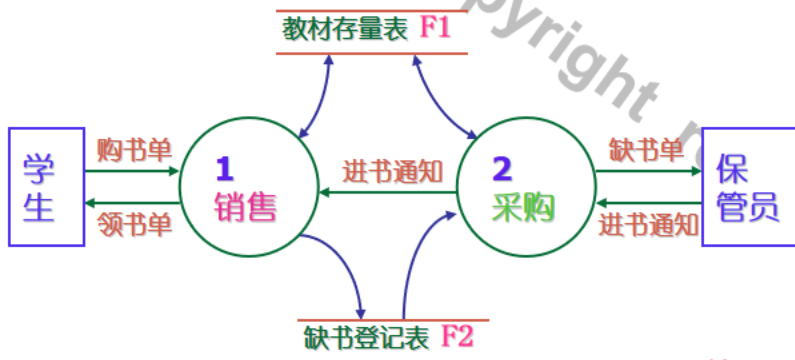


需求分析

学生购书系统---数据流图

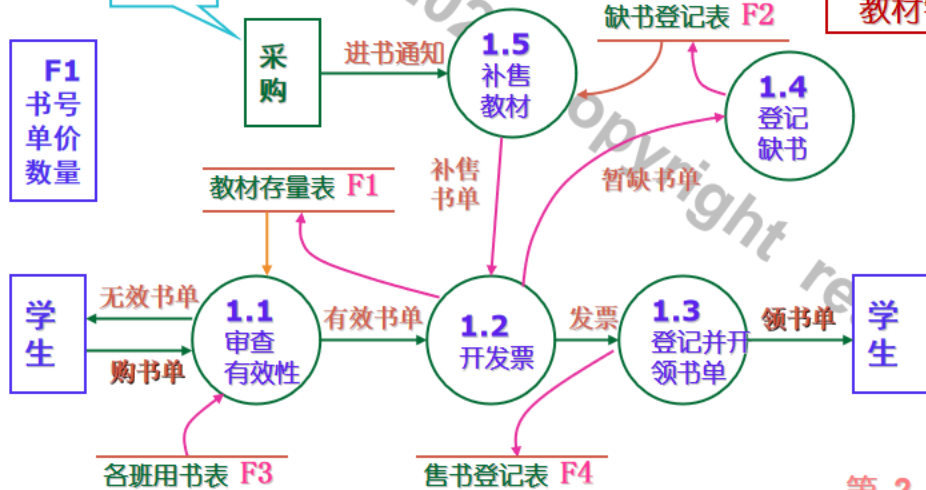
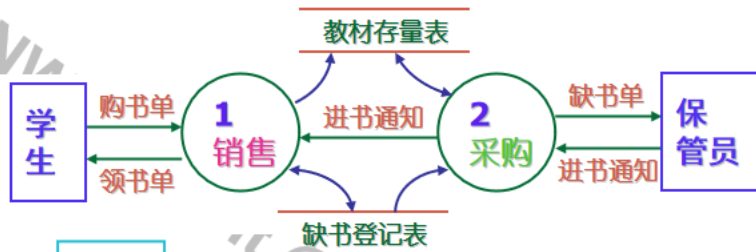


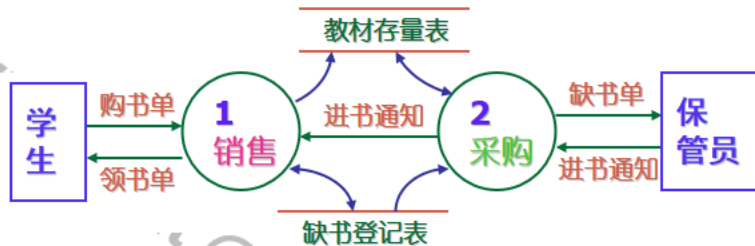
第 0 层



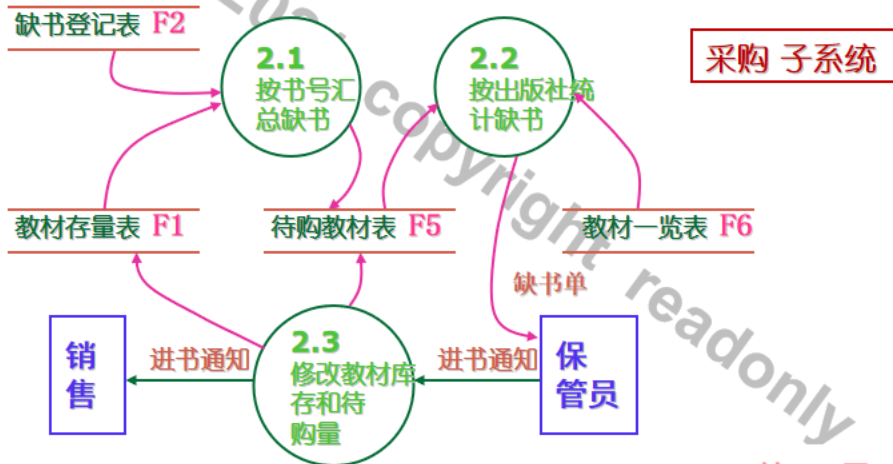
第 1 层







第 1 层



第 2 层



西北工业大学

NORTH WESTERN POLYTECHNICAL UNIVERSITY

行为建模

行为建模给出需求分析方法的所有操作原则，但只有结构化分析方法的扩充版本才提供这种建模的符号。

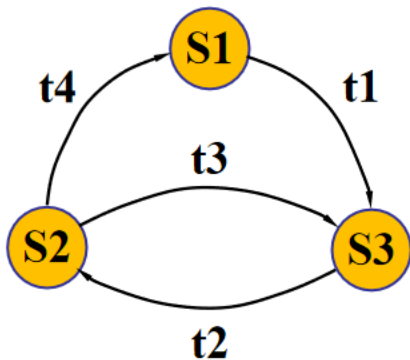
- ❖ 状态迁移图
- ❖ Petri网



状态迁移图

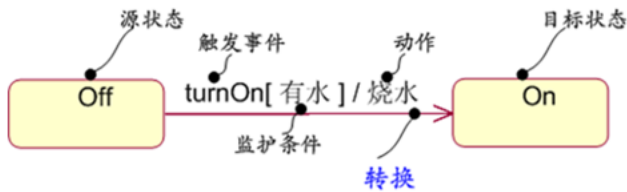
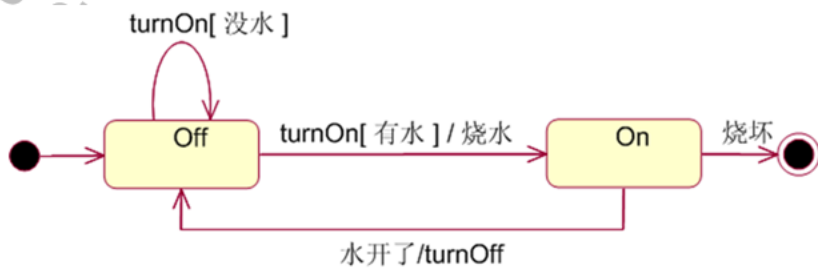
状态迁移图是描述系统的状态如何响应外部的事件进行推移的一种图形表示。

- “○” 表示可得到的系统状态
- “→” 表示从一种状态向另一种状态的迁移。



状态 事件	S1	S2	S3
t1	S3		
t2			S2
t3		S3	
t4		S1	

举例：状态迁移图(电热水壶)



源状态：即受转换影响的状态

目标状态：当转换完成后对象的状态

触发事件：用来为转换定义一个事件，包括调用、改变、信号、时间四类事件

监护条件：布尔表达式，决定是否激活转换、

动作：转换激活时的操作



数据字典

- **定义：**
数据流图中包含的所有元素的定义的集合。
- **用途：**
作为分析阶段的工具。



数据字典与数据流图的联系

- 数据字典定义数据流图中出现的所有的名字。
- 数据流图中出现的**每一个数据流名**，**每一个文件名**和**每一个加工名**在字典中都应有一个条目给出这个名字的定义。
- 数据流图的**每一个组成部分**在字典中也应有一个条目给出它们的定义。



数据字典---数据结构的描述

定义方法： - 顺序 - 选择 - 重复 - 可选

$X = a + b$

$X = [a, b]$

$X = [a|b]$

$X = (a)$

$X = \{a\}$

$X = m\{a\}n$

$X = a ..b$

$X = "a"$

X是由a和b构成

X是由a或b构成

X是由a或b构成

a可在X中出现，也可能不出现

X由零次或多次重复的a构成

X由m至n个a组成，

即至少有m个a，至多有n个a

X可取a至b的任一值

X为取值a的基本数据元素，

即a无需进一步定义



数据定义 例子

户名:	储蓄网点名称:	帐号:	开户日:			
日期(年月日)	摘要	支出	存入	余额	操作	复核
性质:				印密:		

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”

印密 = “0”

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” .. “9999999999.99”

... ..



字典条目类型

数据项(指不再分解的数据单位)

数据项描述={数据项名, 数据项含义说明, 别名, 数据类型, 长度, 取值范围, 取值含义, 与其他数据项的逻辑关系}

数据结构

数据结构描述={数据结构名, 含义说明, 组成:{数据项或数据结构}}

数据流

数据流描述={数据流名, 说明, 数据流来源, 数据流去向, 组成:{数据结构} 平均流量, 高峰期流量}

数据存储/文件

数据存储描述={数据存储名, 说明, 编号, 流入的数据流, 流出的数据流, 组成:{数据结构}, 数据量, 存取方式}

处理/加工

处理过程描述={处理过程名, 说明, 输入:{数据流}, 输出:{数据流}, 处理:{简要说明}} (简要说明常使用工具: 结构化语言、决策表、决策树)



结构化语言

- 结构化语言（如结构化英语）：介于自然语言和形式化语言之间的半形式化语言，是自然语言的一个受限子集。
- 结构化语言没有严格的语法，它的结构通常可分为内外二层。
 - 外层有严格的语法，利用if-then-else, for-do, while-do等关键字分割正文结构。
 - 内层的语法比较灵活，可接近于自然语言的描述，用于描述操作或决策条件。



结构化语言

- 结构化语言的词汇表由
 - 命令动词
 - 数据字典中定义的名字
 - 有限的自定义词
 - 逻辑关系词 if_then_else、switch_do、while_do、for_do、do_while等组成。
- 其基本控制结构有三种：
 - 简单陈述句结构：避免复合语句。
 - 重复结构：while_do、for_do、do_while结构。
 - 判定结构：if_then_else、switch_do 结构。



结构化语言

例：商店业务处理系统中“检查发货单”

```
if 发货单金额超过$500 then
    if 欠款超过了60天 then
        在偿还欠款前不予批准
    else (欠款未超期)
        发批准书，发货单
else (发货单金额未超过$500)
    if 欠款超过60天 then
        发批准书，发货单及赊欠报告
    else (欠款未超期)
        发批准书，发货单
```



判定表

如果数据流图的加工需要依赖于多个逻辑条件的取值，使用判定表来描述比较合适。

条件	条件项	规则
动作	动作项	



判定表举例

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



判定表举例

(另一种画法)

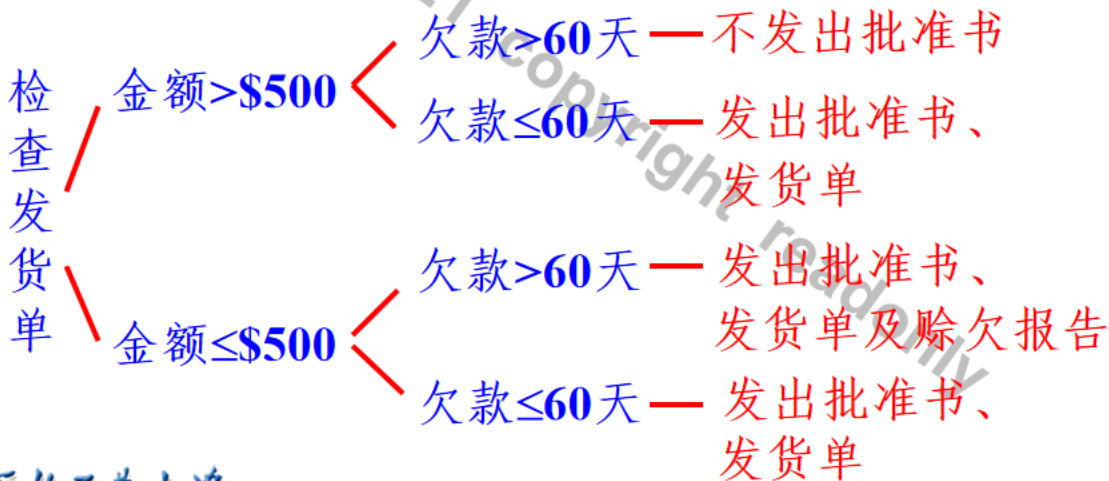
成绩察看条件一览

条件 参看点	员工考生				临时考生			
	立即阅卷(Y)		立即阅卷(N)		立即阅卷(Y)		立即阅卷(N)	
	卷面成绩公开		卷面成绩公开		卷面成绩公开		卷面成绩公开	
	Y	N	Y	N	Y	N	Y	N
客观题成绩	有	有	有	未公开	有	有	有	未公开
主观题成绩	有	未公开	有	未公开	有	未公开	有	未公开
总成绩	有	未公开	有	未公开	有	未公开	有	未公开
成绩概要	有	有(主观题为未公开)	有	未公开	有	有(主观题为未公开)	有	未公开
卷面详细和答案	有	无	有	无	无			

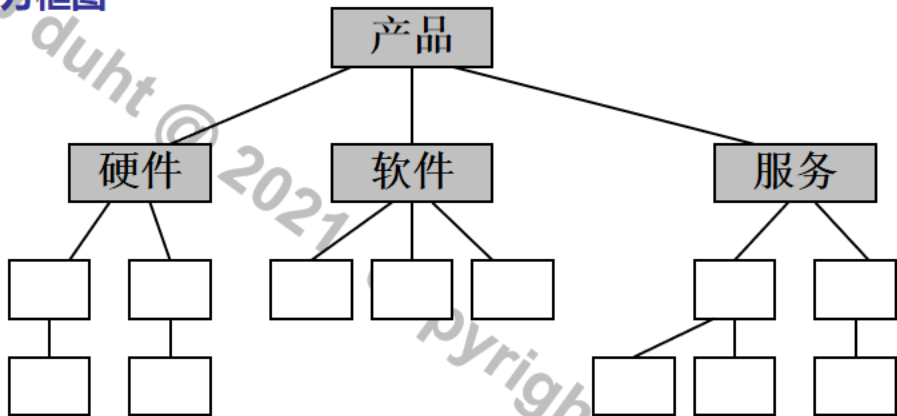


判定树

判定树也是用来表达加工逻辑的一种工具。
有时候它比判定表更直观。



- 层次方框图



- Warnier图

- IPO图

- UML

类图、消息时序图、用例图。。。





需求分析的重要性



软件需求的定义和层次



需求工程



需求分析方法



需求分析文档的原则和评审



需求管理



需求说明书的作用

- 作为用户和软件人员之间的合同，为双方相互了解提供基础
- 反映出问题的结构，可以作为软件人员进行设计和编写的基础
- 作为验收的依据，即作为选取测试用例和进行形式验证的依据

用简单的话来说明需求过程，就是确定系统该做些什么以及该符合什么条件。



软件需求规格说明的内容

- 1) **用户的特点**：列出本软件的最终用户的特点，以及本软件的预期使用频度。这些是软件设计的重要约束。
- 2) **假定和约束**：列出进行本软件开发工作的假定和约束，例如经费限制、开发期限等。
- 3) **需求规定**：
 - ✓ 对功能的规定
 - ✓ 对性能的规定（精度、时间特性、灵活性）
 - ✓ 输入输出要求



- ✓ 数据管理能力要求
- ✓ 故障处理要求
- ✓ 其他如安全保密要求，使用方便要求，可维护性、可补充性、易读性、可靠性、运行环境可转换性等特殊要求等。

4) 运行环境规定

- ✓ 设备
- ✓ 支持软件
- ✓ 接口
- ✓ 控制



1. 介绍

- 1.1 目的
- 1.2 文档结构
- 1.3 预期读者
- 1.4 产品范围
- 1.5 参考资料

2. 总体描述

- 2.1 产品远景
- 2.2 产品功能
- 2.3 用户分类和特征
- 2.4 操作环境
- 2.5 设计和实现约束
- 2.6 用户文档
- 2.7 假设和依赖

3. 外部接口需求

- 3.1 用户接口
- 3.2 硬件接口
- 3.3 软件接口
- 3.4 通讯接口

4. 用例

- 4.1 用例名称1
 - 4.1.1 用例描述
 - 4.1.2 业务规则
 - 4.1.3 功能性需求
- 4.x 用例名称x

5. 非功能需求

- 5.1 性能需求
- 5.2 安全性需求
- 5.3 权限需求
- 5.4 软件质量属性
- 5.5 业务规则（全局）

6. 其他需求

7. 全局功能定义（可选）

附录1. 术语表

附录2. 分析模型

附录3. 已确定的需求列表



如何描述产品或产品构件需求 - 1

- **商业需求**

- 客户或商业利益的描述
 - “提高市场占有率X%”
 - “每年节约电费Y¥”
 - “每年节约维护成本Z¥”

- **商业规则**

- 描述一个方针、标准或规律
 - “只有现场操作人员才能越过安全警告”
 - “必须符合FDA标准的137-B”

- **用例或场景**

- 描述一个用户需要执行的任务
 - “为一个包裹打印邮件标签”
 - “管理需要分析的一系列样本”



如何描述产品或产品构件需求 - 2

- 外部接口需求

- 描述系统与外部世界的关联

- “必须输出一个CSV格式的文件”
 - “必须读取从XXX系统传过来的信号”

- 质量属性

- 描述系统如何很好地执行某些功能

- “系统必须7x24小时提供服务”
 - “复杂数据的响应时间不能超过60秒”

- 约束

- 对设计和实现的限制

- “必须在4M内存上运行”
 - “数据必须向前版本兼容”
 - “必须使用Oracle9.1或之后的运行引擎”



如何描述产品或产品构件需求 - 3

- **数据定义**

- 描述一个数据项或结构

- “费用编号的格式是一个3字节的部门前缀，一个连字号，和一个6位项目编码”
 - “控制温度的缺省值是37度”

- **方案构想**

- 建议一个可能解决问题的途径

- “用户从下拉框中选择期望的值”
 - “用JAVA编写代码”

- **功能需求**

- 表明系统在确定条件下的行为

- “如果压力超过了40，则压力警告灯应该亮”
 - “用户可以按字母顺序排列项目列表”



写作要点

- **简洁**：保持语句和段落的简短；
- **一致**：上下文用语一致，陈述风格应该具有一致的样式。
- **具体**：例如：系统必须…；或者；用户必须…要紧跟着一个行为动作或可观察的结果。举例：计算过程中出现除零错误时，系统必须给出提示信息（**不好！不具体**）； 计算过程中出现除零错误时，系统必须立即弹出对话框显示该错误，并进行声音提示（**好！**）。

※必须避免模糊的、主观的术语，减少不确定性

避免：用户友好、容易、简单、美观、迅速、有效、支持、许多、最新技术、优越的、可接收的和健壮的。

※避免使用比较性词汇，例如：提高、最大化、最小化、最佳化。定量地说明所需要提高的程度或者说清一些参数可接受的最大值和最小值（如“提高投影仪的投影高度”，每个人理解的高度一样嘛？）

※不应该把多个需求集中在一个冗长的叙述段落中。务必记住不要在需求说明中使用“和/或”，“等等”之类的连词（这个等，对于测试人员来说就是灾难...）



编写需求文档的几点建议

- 句子和段落要短，要采用主动语气，使用术语时要保持一致；
- 从客户的角度看看需求是否被有效的定义；
- 从测试人员的角度看看是否便于系统测试用例的编写；
- 正确的把握细化程度，要避免包含多个需求的长的叙述段落；
- 不要使用“也许”，“大概”，“可能”这样的表达模糊性质的词汇；
- 通篇文档细节上要保持一致；
- 多使用图和表；



软件需求规格说明的特征

- 1) **完整**：再也没有什么比软件开发接近完成时才发现遗漏了一项需求更糟的事情了。需求的完整性是非常非常重要的，想象一下遗漏需求而不得不返工，这简直就是恶梦。
- 2) **正确**：文档中的描述是否复合对需求的理解。
- 3) **一致**：需求之间是否有冲突。如果不能同时满足两个需求，则这两个需求就是不一致的。
- 4) **现实**：解决方案确实存在吗？指定的需求以现有软硬件条件是可以支持的吗？
- 5) **无二义**：文档中的描述是否意思准确，多个读者能够一致、有效的理解。
- 6) **可测试、可验证**：需求分析是测试计划的输入和参照。这就要求需求分析是可测试/验证的。



- 完整性

- 需求规格说明应该包括软件要完成的全部任务，不能遗漏任何必要的需求信息。

- 审查需求的完整性应该考虑的问题

- 是否存在遗漏的功能或业务过程？
- 在每个定义的功能之间是否有接口？
- 是否有信息或消息在所定义的功能之间传递？
- 是否定义了功能的使用者？
- 是否已经清楚地定义了用户与功能之间的交互？
- 是否定义了与外部过程和系统之间的接口？



- 正确性

- 需求规格说明对系统功能、行为、性能等的描述必须与用户的期望相吻合，代表了用户的真正需求。

- 审查需求正确性应该考虑的问题

- 用户参与需求过程的程度如何？
- 每一个需求描述是否准确地反映了用户的需要？
- 系统用户是否已经认真考虑了每一项描述？
- 需求可以追溯到来源吗？

- 举例：下面的需求描述正确吗？

- 在用户每次存钱的时候系统将进行信用检查。



- 一致性

- 需求规格说明对各种需求的描述不能存在矛盾，如术语使用冲突、功能和行为特性方面的矛盾以及时序上的不一致等。

- 审查需求的一致性应该考虑的问题

- 文档的组织形式是否易于一致？
- 不同功能的描述之间是否存在矛盾？
- 是否存在有矛盾的需求描述或术语？
- 文档中是否存在时序上的不一致？

- 举例：下面的两个需求描述是否有矛盾？

- 系统允许立即使用所存的资金。
- 只有在手工验证所存资金后，系统才能允许使用。



- 无二义性

- 需求规格说明中的描述对于所有人都只能有一种明确统一的解释。

- 审查需求的无二义性应该考虑的问题

- 需求规格说明是否有术语词汇表？
- 具有多重含义或未知含义的术语是否已经定义？
- 需求描述是否可量化和可验证？
- 每一项需求都有测试准则吗？

- 举例：下面的需求描述是无歧义的吗？

- 如果用户试图透支，系统将采取适当的行动。



• 可验证性

- 需求规格说明中描述的需求都可以运用一些可行的手段对其进行验证和确认。

• 审查需求的可验证性应该考虑的问题

- 在需求文档中是否存在不可验证的陈述，诸如“用户界面友好”、“容易”、“简单”、“快速”、“健壮”、“最新技术”等？
- 所有描述都是具体的和可测量的吗？

• 举例：下面的两个需求描述中哪一个难以验证？

- 系统将在 20 秒内响应所有有效的请求。
- 如果用户试图透支，系统将采取适当的行动。



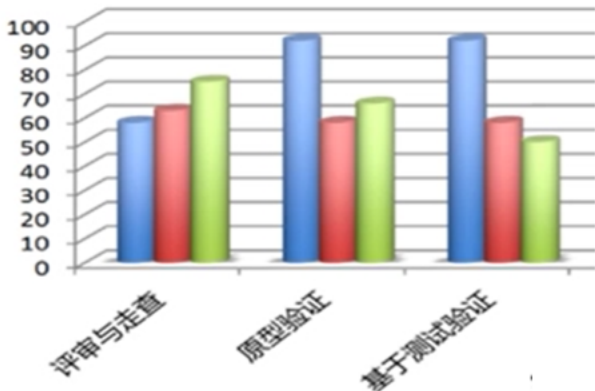
需求验证

- **需求验证**是检验需求能否满足客户的意愿。
- 需求验证的技术
 - **需求评审**：由不同代表（如分析员、客户、设计人员、测试人员）组成的评审小组以会议形式对需求进行系统性分析。
 - **原型评价**：客户和用户在一个可运行的系统模型上实际检验系统是否符合他们的真正需要。
 - **测试用例生成**：通过设计具体的测试方法，发现需求中的许多问题。
- 需求验证主要围绕需求规格说明的质量特性展开。

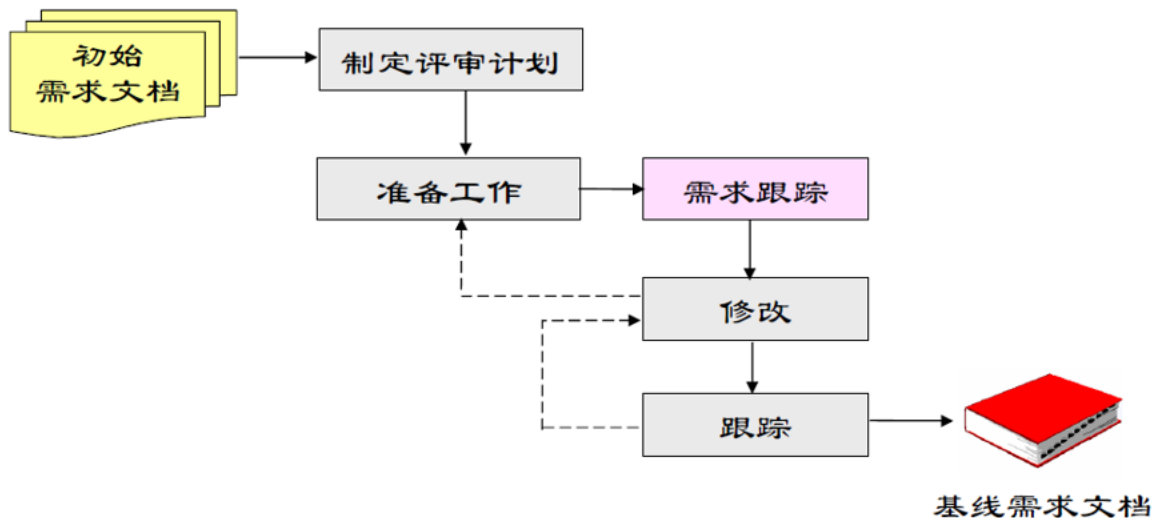


验证技术比较

- ▶ 评审/走查、测试是软件质量保证的重要方法。
- ▶ 原型法在必要时采用原型法进行验证，
- ▶ 3种方法同时作为质量保证-验证需求的方法，互为补充。



评审过程



需求评审

- 1) 系统定义的目标是否与用户的要求一致;
- 2) 系统需求分析阶段提供的文档资料是否齐全;
- 3) 文档中的所有描述是否完整、清晰、准确反映用户要求;
- 4) 与所有其它系统成分的重要接口是否都已经描述;
- 5) 主要功能是否已包括在规定的软件范围之内, 是否都已充分说明;
- 6) 软件的行为和它必须处理的信息、必须完成的功能是否一致;
- 7) 设计的约束条件或限制条件是否符合实际;
- 8) 是否详细制定了检验标准, 它们能否对系统定义是否成功进行确认;





需求分析的重要性



软件需求的定义和层次



需求工程



需求分析方法



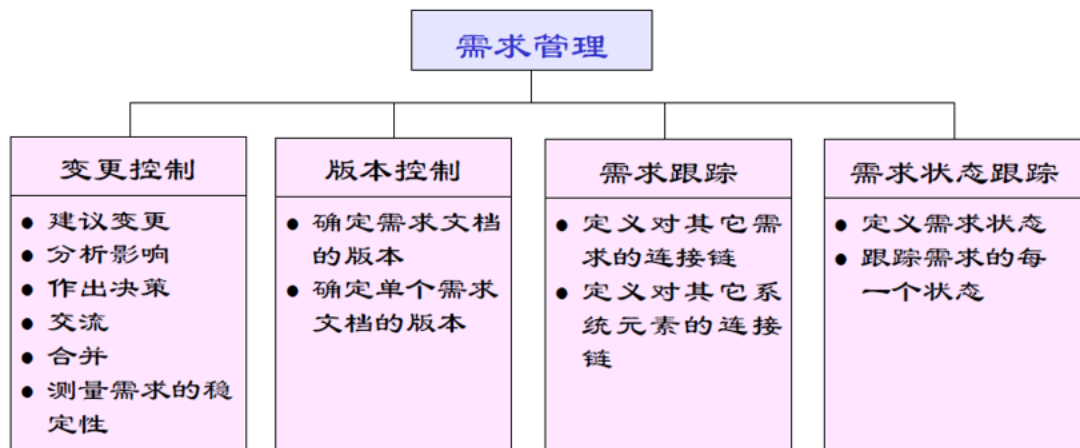
需求分析文档的原则和评审



需求管理



- 需求管理是分析变更影响并控制变更的过程，主要包括变更控制、版本控制和需求跟踪等活动。



需求跟踪矩阵 (Requirement tracking matrix , RTM) 是一种主要管理需求变更和验证需求是否得到了实现的有效工具, 借助RTM, 可以跟踪每个需求的状态。

是否设计了, 是否实现了, 是否测试了。

用例	功能需求	设计元素	代码	测试实例
UC1	Catalog.query.sort	Class catalog	Catalog.sort ()	test2
				test3
UC2	Catalog.update	Class catalog	Catalog.update ()	test10
				test11
				test12



需求分析

需求变更管理例子 (2)

需求ID	需求名	优先级	需求关联	更改次数	状态	来源	SRS的参	概要设计	详细设计	设计变更次数	源程序文件	源程序文件变更次数	测试用例	测试用例变更次数	备注
1	从媒体日志文件采集数据	高	0	代码实现	功能性需求	3.1	4.1	SD4.1 UID2.2.11	变更1次 (SMBSC19)		Begin.jsp Begined.jsp Collection.java SmbsTimerTask.java smbs.sql Closed.jsp	Build1.2 增加 Closed.jsp	SMBS-TC-FT-DC-001~ SMBS-TC-FT-DC-005		
2	构造组合查询, 查看结果	高	0	代码实现	功能性需求	3.2	4.2	SD4.2 UID 2.2.2 UID2.2.10	变更1次 (SMBSC19)		CollectField.java ComboQuery.java DisplayQuery.jsp InputQuery.jsp NoDataFound.jsp ProcessQuery.jsp		SMBS-TC-FT-CQ-001 ~ SMBS-TC-FT-CQ-008		
3	构造交叉统计报表	高	0	代码实现	功能性需求	3.3	4.3	SD4.3 UID2.2.3 UID2.2.9	变更2次 SMBSC20 SMBSC28		CrossTabInput.jsp CrossTable.java DisplayCrTab.jsp DisplayCrXML.jsp	Build2.0 增加 DisplayCrXML S.jsp	SMBS-TC-FT-CS-001~ SMBS-TC-FT-CS-008 SMBS-TC-FT-CS-009		SMBS-TC-FT-CS-009 (变更1次 —新增)
4	计算媒体访问费用	高	0	代码实现	功能性需求	3.4	4.4	SD4.4 UID2.2.16	变更1次 (SMBSC21) 增加 UID2.2.16		computeCost.java Filelength.java Filesize.java Starttime.java countCost.java result.jsp		SMBS-TC-FT-VF-001~ SMBS-TC-FT-VF-003		
5	定义计费策略	高	0	代码实现	功能性需求	3.5	4.5	SD4.5 UID2.2.4 UID2.2.5	变更1次 (SMBSC22)		changeRule1.jsp CountRule.java CountRule1.java define.jsp RuleControl.jsp		SMBS-TC-FT-DF-001~ SMBS-TC-FT-DF-007		
6	用户月结算	高	0	代码实现	功能性需求	3.6	4.6	SD4.6 UID2.2.12	变更1次 (SMBSC23)		DisplayMonthPayment. sp MonthPayInput.jsp Payment.java		SMBS-TC-FT-MC-001~ SMBS-TC-FT-MC-004		
7	用户临时结算	高	0	代码实现	功能性需求	3.7	4.7	SD4.7 UID2.2.13			ProcessTemp.jsp TempPayInput.jsp		SMBS-TC-FT-TC-001~ SMBS-TC-FT-TC-003		

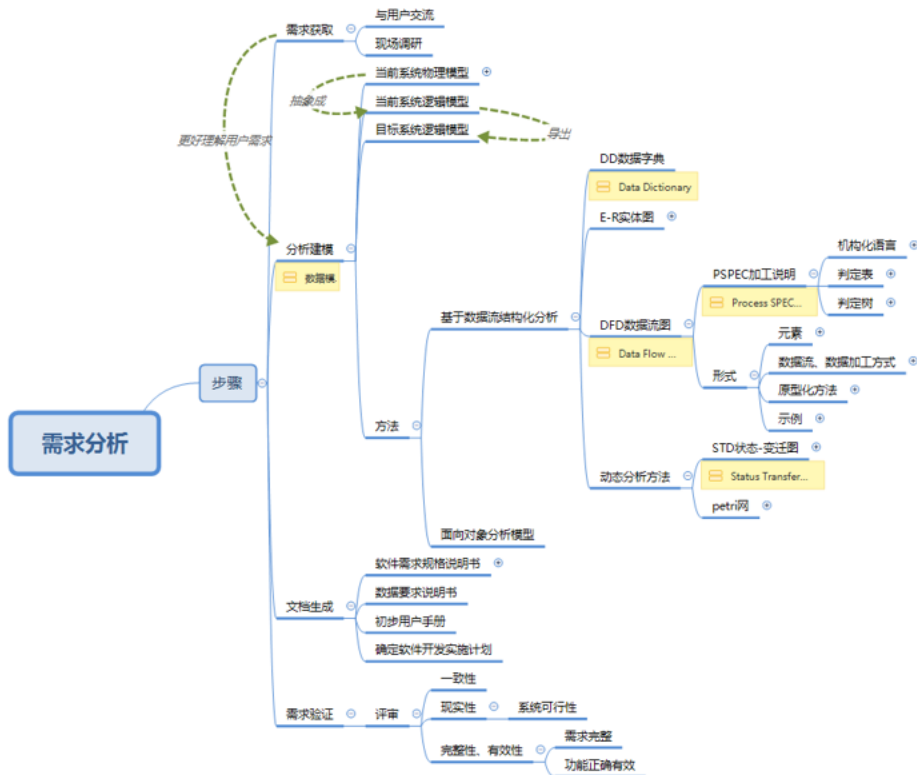


- 1) 需求的内容
- 2) 需求的过程
- 3) 需求分析的方法
- 4) 结构化分析的分析模型
- 5) E-R图
- 6) 数据流图
- 7) 状态迁移图
- 8) 结构化语言

n/y



需求分析思维导图



<https://blog.csdn.net/Elsa15>



西北工业大学

NORTH WESTERN POLYTECHNICAL UNIVERSITY

附录：需求工程师的能力要求

- 分析问题&解决问题的能力
- 人际沟通交流能力
- 软件工程知识和技能（开发经验）
- 应用领域有关知识（业务专业能力）
- 书面语言组织和表达能力
- 其他能力。。。

应避免的问题

- 用户需求=产品需求，不懂挖掘
- 看不起小需求，不注重细节
- 存在即合理，不敢质疑
- 只看重商业价值，不重视用户体验

参考链接：<https://www.aiyingli.com/43085.html>



西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY

下列哪项需求描述属于业务需求描述？

- ☐ A 我们的任务是无缝集成有竞争力的软件信息服务来解决商业问题
- ☐ B 我们的目标是让客户将我们的品牌和质量联系在一起
- ☐ C 我们公司的主营业务是销售飞机票
- ☐ D 公司网站上销售的产品必须满足所有食品药品监管需求

提交



需求管理只适用于需求分析阶段。

- ☐ A 正确
- ☐ B 错误
- ☐ C 说不清楚
- ☐ D 都可以

提交



下面属于需求收集方法的有

A

问卷

B

现场观察

C

客户面谈

D

需求专题讨论会

提交



判断对错。

- ☒ A 非功能性需求就是质量属性需求。
- ☒ B 需求的确认必须由客户或客户方的代表认可。
- ☒ C 需求管理是对变化需求的导出、编档、组织和跟踪。
- ☒ D 客户需求只需要收集功能性需求，其他需求可以不考虑。

提交



需求工程包括哪些基本活动？每一项活动的主要任务是什么？

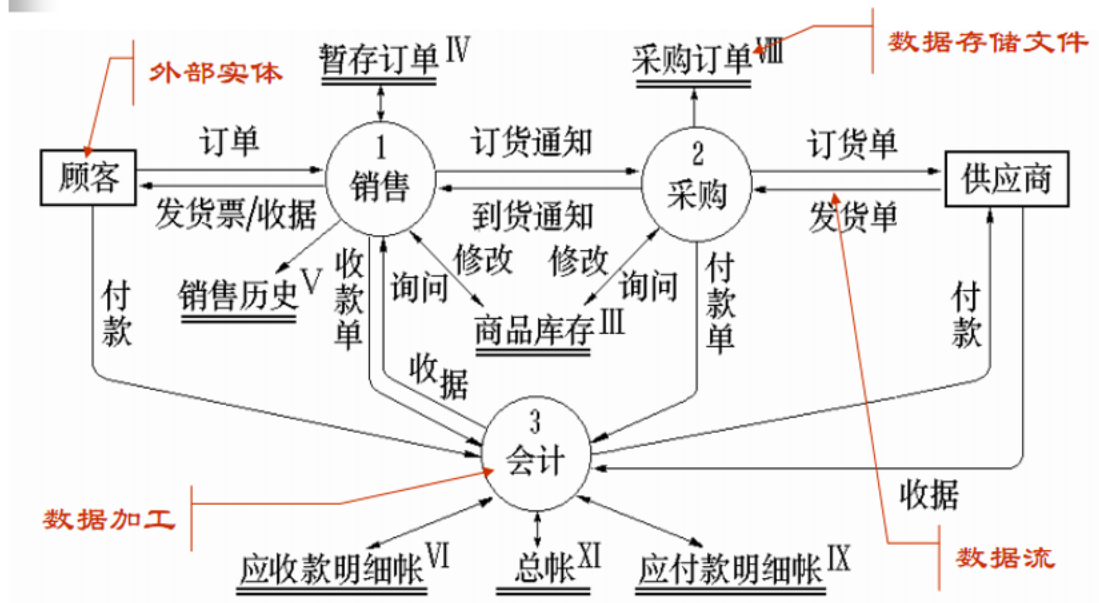


画出**图书预定系统**的数据流图。

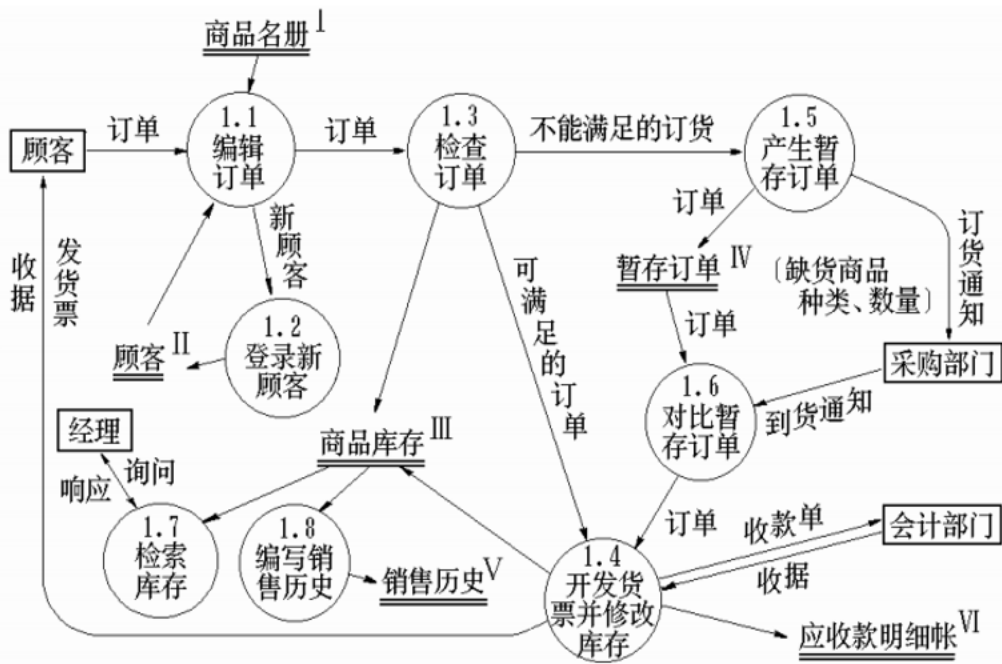
图书预订系统：书店向顾客发放订单，顾客将所填**订单**交由系统处理，系统首先依据图书目录对订单进行**检查**并对**合格订单**进行处理，处理过程中根据顾客情况和订单数目将订单分为**优先订单**与**正常订单**两种，随时处理优先订单，定期处理正常订单。最后系统根据所处理的**订单****汇总**，并按出版社要求发给出版社。



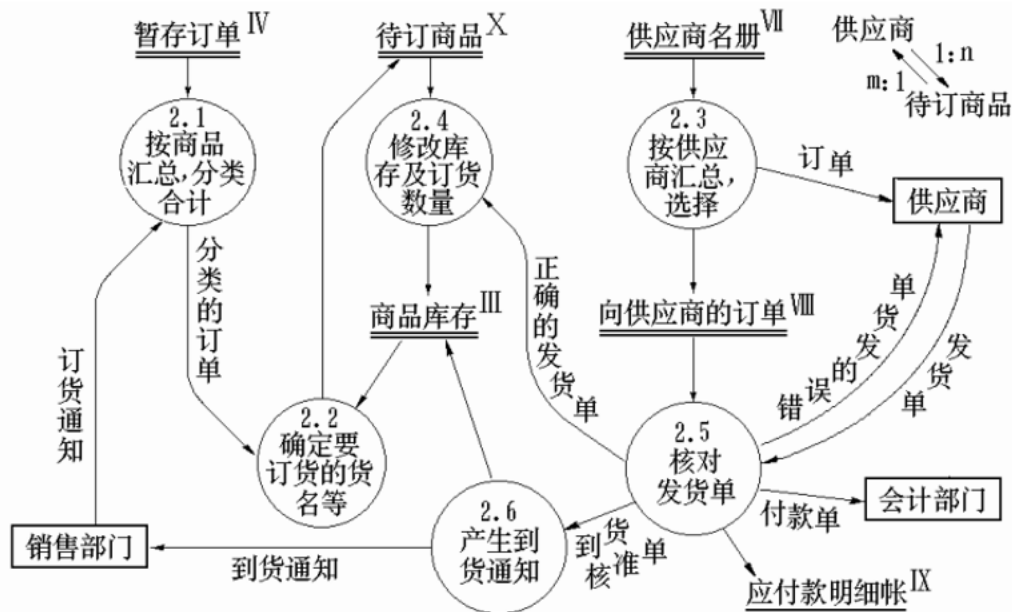
数据流图增加例子：进销存系统



数据流图增加例子：销售子系统



数据流图增加例子：采购子系统





西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY



谢谢!



西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY