

2.1 软件需求的基本概念

软件需求是准确回答“系统必须做什么”的问题

需求分析的基本任务是：

- 定义软件的适用领域和必须满足的约束;
 - 确定系统功能、性能、领域等内容;
 - 确定软件与其他部分间的接口和通信;
 - 建立数据模型、功能模型和行为模型;
- 最终完成需求规格说明, 并经过技术审查和管理复审, 用评价语确认测试和质量评估的依据;

软件需求工程:

- | | | |
|------------------|---|---------|
| 1. 确定系统所要实现的各项需求 | 系统中最重要、最基本的要求是功能需求,是用户/系统中最重要的部分。 | |
| 2. 数据分析 | 将数据从物理实体转换为逻辑模型,分析逻辑数据间的连接和约束模型 | 结构化/开发方 |
| 3. 定义逻辑模型 | 把问题域问题转化为信息域。系统的逻辑模型与软件开发方法有关 | 面向对象/开发 |
| 4. 适应需求变更 | <p>基础线定义之前,能够随时进行需求变更</p> <p>基础线定义之后变更,则需要重新进行审查和评审</p> | |

2.1.2需求分析的原则

- 1.软件人员要从用户的角度考虑软件需求
- 2.以流程为主线
- 3.尽量重用
 - 重用能提高开发效率,节约成本,更重要的是能提高软件质量
- 4.细分需求的优先级
 - 需求分析阶段,重用设计模式是考虑的重点。
- 5.需求变更要及时反馈

2.1.3需求分析的内容

- | | | |
|------|---|--|
| 功能需求 | <p>有时还需说明系统不该做什么。</p> <p>软件的功能描述应该满足完整性和一致性</p> | <p>完整性：需求分析厘清用户提出的所有需求</p> <p>一致性：要求功能需求中不能出现视矛盾之处</p> |
| 性能需求 | 软件系统必须满足时间和空间上的约束 | |
| 领域需求 | 与软件系统的具体应用范围有关，是对需求中的功能或数据在领域上需要的特别是，具有特殊性 | |
| 其他需求 | 与软件系统有关的外在约束，如法律法规、道德需求、外部数据交换需求、预测需求。 | |

2.2可行性分析 (1*)

力求用最小的代价、在尽可能短的时间内确定问题是否能够被解决

-
- ```

graph LR
 A[可行性研究的内容与任务] --> B[经济可行性]
 A --> C[技术可行性]
 A --> D[操作可行性]
 A --> E[法律可行性]
 A --> F[其他]

 B --> B1["回答“系统的经济效益能超过开发成本吗”"]
 B --> B2["运用软件成本预算估计、成本/效益分析等方法"]
 B --> B3["软件成本估计技术"]
 B3 --> B3a[代码行技术]
 B3 --> B3b[功能点技术]

 C --> C1["回答“现有技术能实现这个系统吗”"]
 C --> C2["从问题的复杂性、现有技术、技术所需代价（费用）等方面考虑"]

 D --> D1["回答“系统的操作方式符合用户操作流程和方法”"]

 E --> E1["系统开发过程中是否遵守法律法规"]

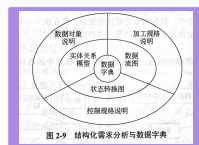
 F --> F1["其他"]

```
- 图 1-1-1 可行性研究的内容与任务图

## 2.3需求工程的过程

求工程在进行可行性研究之后,通过需求获取、需求分析与建模、

- 对需求工程的不断地演进,最终得到公认的需求规格说明。
- 需求工程
- 1.需求获取 采用需求获取技术,得到用自然语言描述的问题的陈述和功能陈述
  - 2.需求分析与建模 以需求获取文档为基础,逐步细化软件功能。
  - 3.需求评审 需求规格说明需要进行技术审查和管理复审
- 需求工程管理两大难题
- 需求确认困难
  - 需求不断变更



## 2.5 结构化分析与建模

结构化需求分析和建模的主要目的是为了减少需求分析时的错误。通过自顶向下建立系统逻辑模型,降低系统设计时的复杂性,提高系统的可维护性。

- 结构化需求分析的核心是数据。数据包括在分析、设计和实现中涉及的概念、术语、属性等所有内容，并把这些内容定义到了数据字典中。

## 面向数据的数据建模

- 数据建模回答软件开发中与各部分设计有关的所有数据对象。数据对象包括实体、实体属性和实体间的关系。
- 实体关系模型（ER图）是结构化建模的可视化图形工具，它描述了数据对象（实体）、对象属性以及对对象间关系和基数。

面向数据流的功能建模

- ```

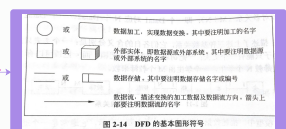
graph LR
    DFD[数据流图 DFD] --- DS[数据源]
    DFD --- DP[数据加工]
    DFD --- DST[数据存储]
    DFD --- DF[数据流]
    DS --- DS_Desc[数据源: 表明数据来源。结果去向。不同层次DFD中, 结果是不能定的]
    DP --- DP_Desc[数据加工: 系统的变换部分, 表明不同数据是通过哪些功能完成的变换]
    DST --- DST_Desc[数据存储: 在DFD中保存数据, 数据结果可以是临时文件, 也可以是持久文件]
    DF --- DF_Desc[数据流: 有方向, 表示数据变换中可以追踪的过程]
  
```

数据流图 (DFD)

 - 数据源**: 表明数据来源。结果去向。不同层次DFD中, 结果是不能定的
 - 数据加工**: 系统的变换部分, 表明不同数据是通过哪些功能完成的变换
 - 数据存储**: 在DFD中保存数据, 数据结果可以是临时文件, 也可以是持久文件
 - 数据流**: 有方向, 表示数据变换中可以追踪的过程

DFD的分解

 - 最上层DFD称为顶层/0层, 又称为逻辑模型, 它反映的是与系统交互的外部系统或用户
 - DFD的分解思想: 自外向内, 逐步向下, 自粗到细, 自顶向下



面向状态转换的行为

- 建模
 - 状态转换图STD
 - 过程描述语言PDL

2.6数据字典 (1*)

据字典是表达系统逻辑模型中所有数据详细情况的工具,它严格的定义了整个系统逻辑模型中所涉及到的基本数据、数据结构、文件形式,每一个元素作为一个词条,在词

- 字典中对该元素的定义、描述、使用特点、与其他数据元素的关系等等都进行了准确的表达。
- 数据的形式定义：表格、范式、Wormior图等形式
- 完整性：在DFD中的数据流名称在DD中必定有相应的名称
 - 一致性：在DFD数据流名与在DD中定义的数据语义必须一致
 - 正确性：数据定义真实的描述了用户领域的数据库、结构、文件等信息。

表 2-2 数据词典定义式中的符号意义

符号	含义	解释
=	被定义为	
+	与	例如, $x=a+b$, 表示 x 由 a 和 b 组成
[...]	或	例如, $x=\{a,b\}$, $y=\{a,b\}$, 表示 x 由 a 或由 b 组成
{...}	或	
[...]	重复	例如, $x=[a]$, 表示 x 由 0 个或多个 a 组成
m[...n]	次数	例如, $x=[a]n$, 表示 x 中至少出现 n 次 a , 至多出现 n 次 a , 也可表示为 $[...n]$
[...]	可选	例如, $x=[a]b$, 表示 a 可在 x 中出现, 也可不出现
*	基本数据元素	例如, $x="a"$, 表示 x 为取值为 a 的数据元素
*	连接	例如, $x=1..9$, 表示 x 可取 1~9 的任一值

表 2-3 用数据表表示数据对象

数据编号	属 性	英文名称	备 注
USER001	姓名	Name	
USER002	所属部门编号	DepartmentID	
USER003	参加工作时间	StartDate	
AUTHORITY	权限	Authority	对应“授权”数据对象
USER004	专业	Major	
USER005	特长	Skill	
RESUME	简历	Resume	对应“简历”数据对象

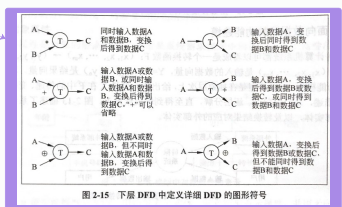


图 2-15 下层 DFD 中定义详细 DFD 的图形符号