

项目三

物流管理信息系统分析

▲ 学习目标

- 了解物流管理信息系统的由来及作用；
- 明确系统分析的流程及关键步骤；
- 掌握信息系统开发的几种常用方法；
- 掌握信息系统分析的流程和系统分析报告的撰写。

▲ 项目描述

本项目介绍物流信息系统开发的方法、物流管理信息系统的分析过程。在物流信息系统开发的方法中介绍了系统开发的相关概念,然后对开发系统的几种常用方法进行讲述,最后介绍了管理信息系统分析的流程。

▲ 专业知识

一、物流管理信息系统开发方法

(一) 系统开发概述

从信息系统的发展和系统特点来看,信息系统可分为数据处理系统(Data Processing System,DPS)、管理信息系统(Management Information System,MIS)、决策支持系统(Decision Sustainment System,DSS)、专家系统(人工智能(AI)的一个子集)和虚拟办公室(Office Automation,OA)五种类型。

信息系统的开发涉及计算机技术基础与运行环境,包括计算机硬件技术、计算机软件技术、计算机网络技术和数据库技术。其中软件分为系统软件和应用软件。系统软件是指为管理、控制和维护计算机及外设,以及提供计算机与用户界面的软件。

管理信息系统的开发原则如下:

(1) 创新原则。计算机技术的发展十分迅速,要及时了解掌握新技术,使目标系统较原系统有质的飞跃。

(2) 整体原则。企业的管理工作可以视为一个闭环系统,开发的目标应当是完善这个闭环系统。

(3) 不断发展原则。为了提高使用率,有效地发挥 MIS 的作用,应当注意技术的发展和环境的变化。

(4) 经济性原则。一个比较合理的 MIS 应与该企业的实际水平挂钩,确保其经济实用性。

(5) 系统原则。管理信息系统作为一个系统,具备系统的基本特性,它可以分解为一组相互关联的子系统,这些子系统各自有其独立的功能,有其边界、输入与输出。

(6) 用户导向原则。管理信息系统的开发包括用户自己,用户不仅是使用管理信息系统的主人,也是开发管理信息系统的主人。只有系统开发人员与用户真诚合作,才是系统成功的关键。

信息系统的开发方法主要包括结构化开发方法、原型法开发方法和面向对象方法等。

(二) 结构化开发方法

结构化开发方法(Structured System Development Methodology)是目前应用得最广泛的一种开发方法,其将开发过程视为一个生命周期,所以又称生命周期法。该方法是用系统的思想和系统工程的方法,按照用户至上的原则结构化、模块化、自顶向下对系统进行分析与设计。结构化开发方法是自顶向下、逐步求精的方法,既符合用户的思维习惯和使用要求,又能根据用户的动态需求调整设计方案,降低了风险,提高了开发效率。其主要特点如下:

(1) 自顶向下分析设计和自底向上逐步实施;

- (2) 深入调查研究和用户至上原则;
- (3) 严格区分工作阶段和开发过程工程化;
- (4) 充分预料可能发生的变化。

结构化开发方法的优点: ①从系统整体出发, 强调在整体优化的条件下“自上而下”地分析和设计, 保证了系统的整体性和目标的一致性; ②遵循用户至上原则; ③严格区分系统开发的阶段性; ④每一阶段的工作成果是下一阶段的依据, 便于系统开发的管理和控制; ⑤文档规范化, 按工程标准建立标准化的文档资料。

结构化开发方法的缺点: ①用户素质或系统分析员和管理者之间的沟通问题; ②开发周期长, 难以适应环境变化; ③结构化程度较低的系统在开发初期难以锁定功能要求。

结构化开发方法主要适用于规模较大、结构化程度较高的系统的开发。

(三) 原型法开发方法

原型就是模型, 而原型系统就是应用系统的模型。与结构化系统开发方法不同, 原型法(Prototyping Approach)开发方法不注重对管理系统进行全面、系统的详细调查与分析, 而是本着系统开发人员对用户需求的理解, 先快速实现一个原型系统, 然后通过反复修改来实现管理信息系统。其开发是一个分析、设计、编程、运行、评价多次重复, 不断演进的过程。

原型法开发方法贯彻的是“从下到上”的开发策略, 它更易被用户接受。原型法开发方法具有如下几方面的特点。

(1) 原型法开发方法自始至终强调用户的参与, 使得用户在系统生存周期的设计阶段起到积极的作用, 扩大用户参与需求分析、初步设计及详细设计等阶段的活动, 加深对系统的理解, 减少系统开发的风险。

(2) 原型法开发方法在用户需求分析、系统功能描述以及系统实现方法等方面有较大的灵活性。特别是在大型项目的开发中, 由于对项目需求的分析难以一次完成, 应用原型法开发方法效果尤为明显。

(3) 原型法开发方法引入了迭代的概念。它既适用于系统的重新开发, 也适用于对系统的修改, 还可以用来建立系统的某个部分。

(4) 原型法开发方法不仅仅局限于对开发项目中的计算机方面进行设计, 第三层原型法开发方法是用于制作系统的工作模型的。快速原型法开发方法要取得成功, 要求有像第四代语言(4GL)这样良好的开发环境与工具的支持。

(5) 原型法开发方法不排斥传统生命周期法中采用的大量行之有效的方法和工具, 它是与传统方法互为补充的方法。

原型法开发方法的优点: ①符合人们认识事物的规律, 系统开发循序渐进, 反复修改, 确保较好的用户满意度; ②开发周期短, 费用相对少; ③由于有用户的直接参与, 系统更加贴近实际; ④易学易用, 减少用户的培训时间; ⑤应变能力强。

原型法开发方法的缺点: ①不适合大规模系统的开发; ②开发过程管理要求高, 整个开发过程要经过“修改—评价—再修改”的多次反复; ③用户过早看到系统原型, 误认为系统就是这个模样, 易使用户失去信心; ④开发人员易将原型取代系统分析; ⑤缺乏规范化的文档资料。

原型法开发方法主要适用于处理过程明确、简单及涉及面窄的小型系统。

(四) 面向对象方法

面向对象方法(Object-Oriented Method)是一种把面向对象的思想应用于软件开发过程中,指导开发活动的系统方法,简称 OO 方法,是建立在“对象”概念基础上的方法学。面向对象方法的基本思想是通过对问题领域进行自然的分割,用更接近人类通常思维的方式建立问题领域的模型,并进行结构和行为模拟,从而使设计出的软件尽可能直接表现出问题的求解过程。

面向对象方法用于开发软件系统时,基于对象(数据)自底向上设计系统结构,从内部结构上模拟客观世界,其主要特点如下:

- (1) 从问题领域的客观事物出发来构造软件系统,用对象作为对这些事物抽象的表示;
- (2) 利用面向对象语言开发软件,实际上就是不断设计和创建类及对象的过程;
- (3) 当一对象发出消息后,由接收对象负责处理,提高了程序执行效率;
- (4) 每一对象的内部状态和功能实现细节对外均是不可见的,提高了程序的可靠性;
- (5) 继承性保证了程序结构紧凑、关系清晰。

面向对象方法的优点: ①分析、设计中的对象和软件中的对象具有一致性; ②实现软件复用,简化程序设计; ③系统易于维护; ④缩短开发周期。

面向对象方法的缺点: 不易于大系统的开发。

就适用范围而言,在大型管理信息系统的开发中,面向对象方法和结构化开发方法相互依存,不可替代。

通过以上分析可见,结构化开发方法、原型法开发方法和面向对象开发方法各有千秋(见表 3-1)。

表 3-1 开发方法的比较

方法名称	开发思想	优点	缺点	适用范围
结构化开发方法	自顶向下	便于系统开发的管理和控制	开发周期较长,环境变化适应性差	适用于规模较大,结构化程度高的系统
原型法开发方法	从模块到整体	用户的直接参与使系统更贴近实际,应变能力强	缺少规范化的文档资料,过程管理要求高	适用于处理过程明确、简单的系统
面向对象开发方法	根据已知进行调查和抽象对象	提高了系统的可维护性和可扩展性	不适合大系统的开发	适用面很广

(五) 系统分析工具简介

系统分析的主要任务是将在系统详细调查中所得到的文档资料集中到一起,对组织内部的整体管理状况和信息处理过程进行分析。它侧重于从业务全过程的角度进行分析。

系统分析的主要内容是业务和数据的流程是否通畅,是否合理;数据、业务过程和实现管理功能之间的关系;老系统管理模式改革和新系统管理方法的实现是否具有可行性等。

系统分析的目的是将用户的需求及其解决方法确定下来。这些需要确定的结果包

括：开发者关于现有组织管理状况的了解，用户对信息系统功能的需求；数据和业务流程，管理功能和管理数据指标体系，新系统拟改动和新增的管理模型等。系统分析所确定的内容是今后系统设计、系统实现的基础。

下面介绍几种系统分析常用工具：判定表(decision table)、决策树(decision tree)、结构化语言(structure language)、数据流图(data flow diagram)。

1. 判定表

1) 判定表的含义

判定表是分析和表达多逻辑条件下执行不同操作情况的工具。判定表又称判断表，是一种呈表格状的图形工具，适用于描述处理判断条件较多，各条件又相互组合、有多种决策方案的情况。精确而简洁描述复杂逻辑的方式，将多个条件与这些条件满足后要执行动作相对应。但不同于传统程序语言中的控制语句，判定表能将多个独立的条件和多个动作直接的联系清晰地表示出来。

判定表的优点是，能够将复杂的问题按照各种可能的情况全部列举出来，简明并避免遗漏。因此，利用判定表能够设计出完整的测试用例集合。

在一些数据处理问题中，某些操作的实施依赖于多个逻辑条件的组合，即针对不同逻辑条件的组合值，分别执行不同的操作。判定表很适合于处理这类问题。

2) 判定表方法

判定表通常由以下四部分组成：条件桩——列出问题的所有条件；条件项——针对条件桩给出的条件列出所有可能的取值；动作桩——列出问题规定的可能采取的操作；动作项——指出在条件项的各组取值情况下应采取的动作。

将任何一个条件组合的特定取值及相应要执行的动作称为一条规则。在判定表中贯穿条件项和动作项的一列就是一条规则。规则的排列方式如表 3-2 所示。

表 3-2 规则排列方式

条件桩	条件项
动作桩	动作项

构造判定表的五个步骤如下：

- (1) 确定规则的个数。有 n 个条件的判定表有 $2n$ 个规则(每个条件取真、假值)；
- (2) 列出所有的条件桩和动作桩；
- (3) 填入条件项；
- (4) 填入动作项，得到初始判定表；
- (5) 简化判定表，合并相似规则。

若表中有两条以上规则具有相同的动作，并且在条件项之间存在极为相似的关系，便可以合并。合并后的条件项用符号“—”表示，说明执行的动作与该条件的取值无关，称为无关条件。

2. 决策树

1) 决策树的含义

决策树一般都是自上而下生成的。每个决策或事件(即自然状态)都可能引出两个或

多个事件,导致不同的结果,把这种决策分支画成图形很像一棵树的枝干,故称决策树。

2) 决策树方法

(1) 选择分割的方法。选择分割的方法有几种,但是目的都是一致的:对目标类尝试进行最佳的分割。从根到叶子节点都有一条路径,这条路径就是一条“规则”。决策树可以是两叉的,也可以是多叉的。对每个节点的衡量方法:通过该节点的记录数;如果是叶子节点的话,通过分类的路径;通过对叶子节点正确分类的比例。有些规则的效果可以比其他的一些规则要好。

(2) 决策树的构成要素。决策树的构成有四个要素:决策结点、方案枝、状态结点、概率枝。

决策树四个要素的构成方式如图 3-1 所示。

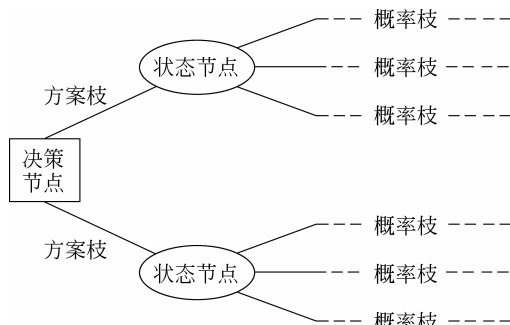


图 3-1 决策树四个要素的构成方式

总之,决策树一般由方块节点、圆形节点、方案枝、概率枝等组成。方块节点称为决策节点,由节点引出若干条细支,每条细支代表一个方案,称为方案枝;圆形节点称为状态节点,由状态节点引出若干条细支,表示不同的自然状态,称为概率枝。每条概率枝代表一种自然状态。在每条细枝上标明客观状态的内容和其出现概率。在概率枝的最末梢标明该方案在该自然状态下所达到的结果(收益值或损失值)。这样树形图由左向右,由简到繁展开,组成一个树状网络图。

3. 结构化语言

结构化语言是受结构化程序设计思想启发而扩展出来的。结构化程序设计只允许采用三种基本结构。结构化语言也只允许采用三种基本语句,即简单的祈使语句、判断语句和循环语句。与程序设计语言的差别在于,结构化语言没有严格的语法规定。与自然语言的不同在于,它只有极其有限的词汇和语句。结构化语言使用三类词汇:祈使句中的动词、数据字典中定义的名词以及某些逻辑表达式中的保留字。

1) 祈使语句

祈使语句指出要做什么事情,包括一个动词和一个宾语。动词指出要执行的功能,宾语表示动作的对象,例如:计算工资、发补考通知。

使用祈使语句时,应注意以下几点。

- (1) 力求精练,不应太长。
- (2) 不使用形容词和副词。

(3) 动词要能明确表达执行的动作,不用“做”、“处理”这类意义太泛的动词;意义相同的动词,只确定使用其中之一。

(4) 名词必须在数据字典中有定义。

2) 判断语句

判断语句类似结构化程序设计中的判断结构,其一般形式如下:

```

如果 条件
    则 动作 A
否则 (条件不成立)
    动作 B
  
```

判断语句中的“如果”、“否则”要成对出现,以避免多重判断嵌套时产生二义性外,书写时每层要对齐,以便阅读。

3) 循环语句

循环语句表达在某种条件下,重复执行相同的动作,直到这个条件不成立为止。

4. 数据流图

1) 数据流图的含义

数据流图(DFD)是一种能全面地描述信息系统逻辑模型的主要工具,它可以用少数几种符号综合地反映出信息在系统中的流动、处理和存储情况。数据流图是一种图形化技术,是系统逻辑功能的图形表示,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,它是分析员与用户之间极好的通信工具,也是分析和设计的工具。

DFD由数据流、加工、数据存储和外部实体四个要素构成。外部实体是指存在于软件系统之外的人员或组织,它指出系统所需数据的发源地和系统所产生数据的归宿地。

2) 数据流图的几种符号

数据流图有以下四种基本图形符号。

→: 箭头,表示数据流。

○: 圆或椭圆,表示加工。

═: 双杠,表示数据存储。

□: 方框,表示数据的源点或终点。

(1) 数据流。数据流是数据在系统内传播的路径,因此由一组成分固定的数据组成。如订票单由旅客姓名、年龄、单位、身份证号、日期、目的地等数据项组成。由于数据流是流动中的数据,所以必须有流向,除了与数据存储之间的数据流不用命名外,数据流应该用名词或名词短语命名。

(2) 加工(又称为数据处理)。加工对数据流进行某些操作或变换。每个加工也要有名字,通常是动词短语,简明地描述完成什么加工。在分层的数据流图中,加工还应编号。

(3) 数据存储(又称为文件),是指暂时保存的数据,它可以是数据库文件或任何形式的数据库组织。

(4) 数据源点或终点,是软件系统外部环境中的实体(包括人员、组织或其他软件系统),统称外部实体。它一般只出现在数据流图的顶层图中。

二、系统分析

(一) 系统详细调查

系统详细调查的目的在于通过对现行系统进行全面、充分和详细的调查,完整掌握现行系统的现状,发现问题和薄弱环节,收集资料,为下一步的系统化分析和新系统的逻辑设计做好准备。

系统详细调查的主要内容包括:管理业务调查和数据流程调查。其中管理业务调查由组织结构调查、管理功能调查和业务流程调查构成。

1. 详细调查的原则

(1) 真实性。所谓真实性是指系统调查资料真实、准确地反映现行系统状况,不依照调查者的意愿反映系统的优点或不足。

(2) 全面性。任何系统都是由许多子系统有机地结合在一起而实现的。

(3) 规范性。有一套循序渐进、逐层深入的调查步骤和层次分明、通俗易懂的规范化逻辑模型描述方法。

(4) 启发性。需要调查人员逐步引导,不断启发,尤其在考虑计算机处理的特殊性而进行的专门调查中,更应该善于按使用者能够理解的方式提出问题,打开使用者的思路。

2. 详细调查的内容

(1) 系统的定性调查。定性调查主要是对现有系统的功能进行总结,包括组织结构的调查、管理功能的调查、工作流程的调查、处理特点的调查与系统运行的调查等。

(2) 系统的定量调查。定量调查的目的是弄清数据流量的大小、时间分布、发生频率,掌握系统的信息特征,据此确定系统规模,估计系统建设工作量,为下一阶段的系统设计提供科学依据。

3. 详细调查的方法

调查的方法多种多样,经常使用的方法如下:

(1) 问卷调查法。可以用来调查系统普遍性的问题,由初步调查结果可得到组织的基本情况。

(2) 召开调查会。这是一种集中调查的方法,适合于了解宏观情况。

(3) 调查人员直接参加业务实践。开发人员亲自参加业务实践,不仅可以获得第一手资料,而且便于开发人员和业务人员的交流,使系统的开发工作接近用户,用户更了解新系统。

(4) 查阅企业的有关资料。

(5) 个别访问。针对某些特殊问题或细节进行调查,可对有关的业务人员作专题访问,仔细了解每一步骤、方法等细节。

(6) 由用户的管理人员向开发者介绍情况。

(7) 其他还有专家调查等方法,可以根据系统调查的具体需要确定调查方法。

（二）系统可行性分析

系统可行性分析的目的在于,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。可行性分析主要是评估问题是否值得解决,这需要客观分析。必须分析几种主要可能解法的利弊,从而判断原定的系统目标和规模是否现实,系统完成后能带来的效益是否大到值得投资开发这个系统的程度。

为了达到避免盲目投资,减少不必要的损失的目的,信息系统的可行性分析一般可从以下四个方面着手。

(1) 技术上的可行性。根据系统的目标来考虑系统的软硬件设备、计算机联网能力、网络及数据安全保护设施、输入输出设备、大容量存储设备等。软件方面应重点考虑操作系统、数据库管理系统等配置和功能。技术力量则考虑技术人员经验和水平。在进行技术可行性分析时,还必须考虑该单位业务人员的文化素质,经过培训后是否能够承担使用和维护工作的可能性。

(2) 经济上的可行性。系统的经济可行性分析是指估算系统开发和运行所需的费用,以及系统的效益,将投资和效益进行比较,说明在经济上是合算的。它包括设备费用、人员费用、材料费用和其他费用等。

(3) 管理上的可行性。考虑当前系统的管理体制是否有条件提供系统所必需的各种数据。

(4) 开发环境的可行性。企业领导的意见是否一致,有无资金,能否抽出骨干力量参加系统开发等。简单地说,就是企业能否为新系统的开发建设提供一个长期良好的环境,这是可行性阶段必须考虑的问题。

系统是企业仓储管理的开发者开发的,使用对象是企业仓库的工作人员。由于系统界面设计友好,有详细的使用说明,用户只需要懂得简单的计算机操作知识,就能应用系统,因此系统完全可行。

（三）系统功能需求分析

1. 需求分析概述

需求分析是一个项目的开端,也是项目建设的基石。在以往建设失败的项目中,有80%是由于需求分析不明确而造成的,因此一个项目成功的关键因素之一,就是对需求分析的准确把握程度。需求分析的定义包括从用户角度(系统的外部行为),以及从开发者角度(一些内部性)来阐述需求,其任务简单地说就是做什么。

整个需求工程可以划分为需求开发和需求管理两部分。

需求开发可进一步分为问题的获取、分析以及编写规格说明和验证四个阶段。需求开发活动主要包括这几个方面,即确定产品所期望的用户类别,获取每个用户类的需求,了解实际用户的任务和目标以及这些任务所支持的业务需求。分析来源于用户的信息,可准确把握用户任务需求、功能需求、业务规则、质量属性、建议解决方法和附加信息等。

需求管理需要建立并维护在软件工程中同客户达成的协议,这一协议包含在编写的需求文档与模型中。客户的接受仅是需求成功的一半,开发人员也必须能够接受它们,并

真正把需求应用到产品中。通常的需求管理活动包括:定义需求基线(迅速制定需求文档的主体),评审提出的需求变更、评估每项变更的可能影响从而决定是否实施。

2. 需求的类型

(1) 业务需求。反映了组织机构或客户对系统、产品较高层次的目标要求,以及它们在项目视图与范围文档中的详细说明。

(2) 用户需求。需求分析文档描述了用户使用产品必须要完成的任务,这在使用实例文档或方案脚本中予以说明。

(3) 功能需求。定义了开发人员必须实现的软件功能,以满足客户需求,使用户能完成他们的任务,从而满足业务需求。

3. 需求分析方法

(1) 自上而下(Top-Down)与自下而上(Bottom-Up)。自上而下,从物流企业管理的整体进行设计,从最顶层开始,逐层深入,直至清楚所有物流管理工作,进而从概要设计到详细设计,体现结构化的设计思想。自下而上,即采用搭积木的方式组成整个系统,缺点在于若考虑不周全可能忽视系统各个部件的有机联系。因此,两者结合是实际开发过程中常用的方法。

(2) 先分析后改进。物流企业内部的每一个管理部门和管理工作都是根据物流组织的具体情况和需要设置的,因此首先必须弄清楚这些管理工作存在的必要性、存在环境以及处理的详细过程;然后分析其在新的物流管理信息系统中有没有优化或者改进的可能。

(3) 全面与重点相结合。对于整个物流管理信息系统而言,开展全面详尽的调查工作是相当必要的。倘若某一时段只开发局部的物流管理信息系统(如仓储管理信息系统),则必须使用全面与重点相结合的方法。

4. 需求跟踪

需求跟踪是需求分析管理的一项重要内容,其主要意义在于获得需求目前的实现状态,确保用户的所有需求都得到满足。需求跟踪是跟踪一个需求使用期限的全过程,包括编制每个需求同系统元素之间联系的文档,这些元素包括体系结构、其他设计部件、源代码模块、测试、帮助文件等。需求跟踪提供了一种由需求到产品整个实现过程范围的明确查阅方法,其目的是建立与维护“需求—设计—编程—测试”之间的一致性,确保所有的工作成果符合用户需求。

需求跟踪有两种方式:正向跟踪,检查《产品需求规格说明书》中的每个需求是否都能在后继工作成果中找到对应点;逆向跟踪,检查设计文档、代码、测试用例等工作成果是否都能在《产品需求规格说明书》中找到出处。

图 3-2 说明了四类需求跟踪能力链。客户需求可向前追溯到需求,这样就能区分出开发过程中或开发结束后由于需求变更受到影响的需求。同样,可以从需求回溯相应的客户需求,确认每个软件需

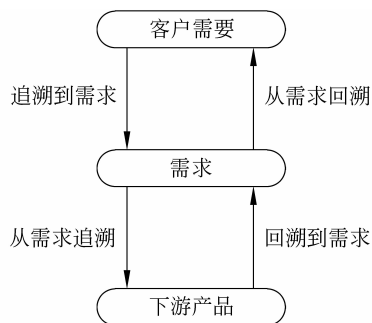


图 3-2 四类需求跟踪能力链