

第3章 信息系统开发方法

信息系统开发的效率、质量、成本以及用户的满意程度,除了管理、技术等因素外,系统开发方法起着极其重要的作用。信息系统开发方法是用已定义的工具、方法和过程,在技术上和管理上来组织软件生产的一系列活动。其中,过程定义了信息系统开发的操作流程,方法是用信息系统开发理论和规范的技术手段进行设计,工具提供了方法中可用的一组图形符号,这些图形符号有各自的语法和语义信息。

信息系统开发方法有多种,常用的开发方法有结构化系统开发方法、面向对象系统开发方法和计算机辅助软件工程方法,它们各自遵循着一定的基本思想,适用于一定的范围,解决问题的出发点和侧重点也各不相同。系统开发方法需要技术上的支持,通常某种方法学是几种技术的有机结合。无论采用何种开发技术与方法,都必须实现两个基本的目标:一是提高系统开发的效率,二是提高系统的质量。

随着技术的进步、管理工作要求的提高,信息系统的开发技术与方法也在不断地丰富和完善,新的开发技术与方法会不断出现。

3.1 结构化系统开发方法

20世纪70年代,西方发达国家在不断的实践中,吸取了以往系统开发的经验和教训,提出了系统结构化分析与设计的方法,即结构化系统开发方法(Structured System Development Method,SSDM)。它是自顶向下的开发策略、工程化的系统开发方法和生命周期方法的结合,是迄今为止的开发方法中最传统、应用最广泛的一种开发方法。

3.1.1 基本思想

结构化系统开发方法是用系统工程的思想 and 工程化的方法,按照用户至上的原则,结构化、模块化、自顶向下地分析和设计系统。在整个开发过程中,把软件生命周期的全过程依次划分为若干个阶段,然后顺序地完成每个阶段的任务,各个阶段和步骤清晰规范,每一阶段和步骤均有明确的内容、成果,阶段成果都以系统规划报告、系统分析报告、系统设计报告、系统实施报告等形式经过技术和管理审查后确定下来,每一阶段的报告都是下一阶段的重要参考依据,各阶段衔接紧密,特别注重整体性和全局性。

3.1.2 过程

基于系统开发生命周期,结构化系统开发方法将信息系统开发过程分为信息系统规划、信息系统分析、信息系统设计、信息系统实施、信息系统运行与维护5个阶段,每一阶段都要严格把关,只有审核通过才能进入下一个阶段。结构化系统开发方法过程如图3-1所示。

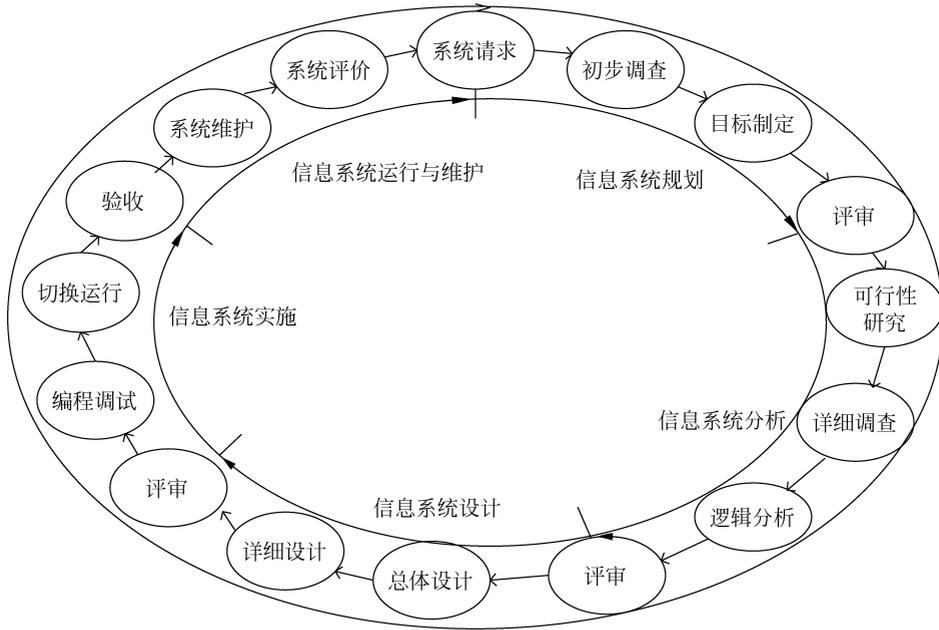


图 3-1 结构化系统开发方法过程

1) 信息系统规划

在这一阶段要根据用户的开发请求进行初步调查,明确问题,制定与组织发展战略相适应的信息系统发展战略,根据组织的主要信息需求形成信息系统的总体结构方案,了解约束条件和所需的基本资源。系统规划主要包括制订系统建设的资源分配计划和项目开发计划。

2) 信息系统分析

系统分析阶段的主要工作是根据系统规划阶段确定系统总体方案和开发项目的安排,对系统进行可行性分析、组织结构分析、业务流程分析、系统数据流程分析、数据字典创建,最后完成新系统的逻辑方案设计,形成系统分析报告等内容。

3) 信息系统设计

系统设计的主要任务是根据新系统的逻辑方案进行软件、硬件系统的设计,包括体系结构设计、系统功能结构设计、数据库设计、编码设计、输入/输出设计、处理过程设计,最后编写系统设计说明书。

4) 信息系统实施

系统实施的主要任务是将设计的系统付诸实施,主要工作有系统开发的组织与管理,应用程序编制或软件包的购置,计算机与通信设备的购置,系统的安装、调试和测试,新系统和现有系统的切换、评估等。

5) 信息系统运行与维护

信息系统开发完成后投入应用,即进入运行和维护阶段。在本阶段,开发者和用户一起支持信息系统的运行,而科学的组织与管理是系统正常运行、充分发挥其作用的必要条件,及时完善的系统维护是系统正常运行的基本保证。信息系统规模庞大,结构复杂,管理环境

和技术环境不断变化,系统维护工作量大,涉及面广,投入资源多。

现代组织面临的内、外环境不断变化,组织的目标、战略和信息需求也必须与环境的变化相适应。当现有系统或系统的某些主要部分已经不能通过维护来适应变化时,或者用维护的办法在现有系统上进行调整的成本过高时,整个信息系统或某个子系统就要被淘汰,新的系统建设工作或项目开发工作便随之开始。

现有系统进入更新阶段时,新系统的建设工作就开始了。因此,这一阶段是新系统和现有系统并存的时期。对现有系统来说,可以全部更新,也可以部分更新或有步骤地分期分批更新。

3.1.3 原则

结构化系统开发方法主要包括如下原则。

1) 用户积极参与

用户是信息系统需求的提供者,也是最终使用者,用户积极参与信息系统开发的全过程,是信息系统开发成功的关键因素。尽量吸引用户全程参与开发过程,加强与用户的联系,以便统一认识,加快工作进度,提高系统质量,减少系统开发的盲目性和失败的可能性。

2) 严格划分阶段

整个信息系统的开发采用“分而治之”的策略,即将系统开发过程分为若干个“阶段”,每个阶段都有明确的任务和目标,各个阶段又可分为若干工作和步骤,逐一完成任务,从而实现预期目标。各个阶段的开发有条不紊地进行,便于计划和控制,避免为以后的工作留下隐患。

3) 设立检查点

在系统开发的每一个阶段均设立检查点,以评估所开发系统的可行性,避免由于系统开发的失败造成更大的损失。

4) 文档资料标准化

根据系统工程的思想,信息系统各个阶段性的成果必须文档化,只有这样才能更好地实现用户与系统开发人员的交流,保证各阶段工作的衔接和维护的便利,为提高信息系统可靠性提供保证。

3.1.4 特点

1. 主要优点

结构化系统开发方法主要具有以下优点。

1) 自顶向下和自底向上相结合的开发过程

在系统分析与设计时要从整体全局考虑,采用自顶向下、逐步分解细化和模块化的思路,系统实施时,根据设计的要求从具体的功能模块入手,自底向上逐步实现整个系统,便于系统的分析、设计、实施和维护,使复杂的系统开发工作简单化。

2) 结构化和模块化

从全局观点出发进行系统分析和设计,保证系统总体结构的合理性、系统内信息的完整

性与一致性、各子系统之间的有机联系。同时,根据设计的要求,采用模块化技术设计、编程与调试,逐步实现整个系统,强调系统开发的整体性和全局性。

3) 严格划分系统阶段

把系统完整的开发过程划分为若干阶段,每一阶段对应的目标和任务明确,前后衔接。在实际的开发过程中,要求严格按照划分的工作阶段逐步开展工作,同时进行严格审查,审查通过后才进行下一阶段工作,使错误较难传递到下一阶段,这在一定程度上减少了错误造成的损失,保证了系统开发的质量。

4) 用户至上

用户对系统开发的成败是至关重要的,所以在系统开发过程中要面向用户,始终保持与用户的沟通,充分了解用户的需求和愿望,详细地调查研究,弄清实际业务处理过程的每一个细节,取得用户的共识,这是信息系统开发的基础。然后分析研究,制订出科学合理的新系统设计方案。

5) 充分预料可能发生的变化

系统开发是一项耗费人力、财力、物力且周期很长的工作,一旦周围环境(组织的内外部环境、信息处理模式、用户需求等)发生变化,都会直接影响到系统的开发工作,所以结构化开发方法强调在系统调查和分析时对将来可能发生的变化给予充分的重视,强调所设计的系统对环境的变化具有一定的适应能力。

6) 文档资料规范化、标准化

系统开发过程工程化,要求开发过程的每一阶段都按工程标准规范化,将文档资料标准化,即采用标准化、规范化的格式和术语、图表等形式组织文档,便于系统开发人员和用户的交流。

2. 主要缺点

1) 必须事先明确需求

结构化方法要求预先严格定义出完整、准确的功能需求和规格说明,为此在早期调查中就要充分地掌握用户需求、管理状况,并能预见将要发生的变化,这不太符合人们循序渐进地认识事务的客观规律。

2) 开发周期长

必须充分了解用户的需求后,才能进行信息系统的开发,系统开发过程中的每一阶段都要和用户充分沟通后,形成规范化、标准化的文档资料,并进行评审,使得系统开发周期过长而带来一系列问题。

3) 灵活性差

开发环境在不断变化,而结构化方法需要按照确定的设计目标进行,难以用于运行环境经常变化的信息系统的开发。

3.2 面向对象方法

按照信息系统分析的要素,可以将信息系统分为面向处理、面向数据和面向对象三种方法。面向处理(Processing Oriented, PO)的方法中常常由企业运营流程出发,划分不同的过

程进行处理分析,此时也称为面向过程的程序设计方法;面向数据(Data Oriented,DO)的方法是分析企业的信息需求,通过信息建模,建立企业的数据库。这两种方法把相互依赖的数据和对数据的操作分离,要么只单纯地反映管理功能的结构状况,要么只侧重反映事物的信息特征和信息流程,使得大型系统难于编写和调试,程序员之间很难读懂对方的代码,更谈不上代码的重用。面向对象(Object Oriented,OO)系统开发方法是基于面向对象的程序设计方法逐步发展的,它把数据和过程包装成为对象,以对象为基础对信息系统进行处理,因此它是一种综合性的开发方法。

3.2.1 基本思想

客观世界中任何具体的事物反映在人的头脑中都是抽象的概念,每一个事物映射到计算机世界中就是面向对象系统开发方法中的一个对象(Object)。面向对象方法认为:客观世界是由各种对象组成的,任何事物都是对象,复杂的对象可以由比较简单的对象以某种方式组合而成,因而信息系统可以抽象为由各种层次、具有各种属性、彼此相互联系又相互作用的一系列对象构成的一个复杂的体系结构。对象是用面向对象开发方法分析问题和解决问题的核心,面向对象的分析过程就是认识客观世界的过程,而且面向对象方法提供了一些支持复用的机制(如继承),这提高了软件系统的可复用性。

3.2.2 面向对象与面向过程

当问题的规模不大时,面向过程的设计方法因其逻辑关系明确、实现简单而备受程序员们的青睐。但是当问题规模扩大到大中型软件时,一个项目的代码量不是一个人可以单独完成的。小组的分工与合作成为解决代码量大的方法,但是这又向程序员之间的配合提出了新的挑战,实时数据共用导致的不安全、代码重用率低等问题也阻碍着面向过程的方法在大型软件中的发展。

面向过程是以事件为中心,分析解决问题所需要的步骤,用函数逐步实现,使用的时候依次调用。而面向对象体现了以事物为中心的编程思想,以对象为基础,以事件或消息来驱动对象执行处理,将数据和对数据的操作封装在一起,并将它们作为一个整体来处理,同时采用数据抽象和信息隐蔽技术,属性和操作相同的对象抽象为类,并考虑类之间的联系和类的重用性。在面向对象程序设计里,一切操作都是通过向对象发送消息来实现的。对象接到消息后,启动消息处理函数完成相应的操作。因此,面向对象程序设计的控制流程是由运行时各种事件的实际发生来触发的,而不再由预定顺序来决定,这更符合实际。

3.2.3 基本概念

1. 对象

对象是任何被人类研究的事物,是建立面向对象系统的基本单元。不管是简单的实数,还是结构复杂的火箭、飞船,甚至抽象的计划,都可以认为是对象。对象是一组属性和施加在这些属性上的一组方法构成的独立个体。对象的属性又称为数据部分,用来描述它的静态特征,对象的方法又称为程序部分,通常是一段程序代码,定义了一个操作,每个操作决定

对象的一种功能或行为,用来描述它的动态特征。以汽车为例,它拥有的发动机,变速箱,车身高、宽、高,重量等是对象的属性,而启动、加速、刹车等是它的方法。

对象是一个封闭体,它向外界提供一组接口,外界通过这组接口与对象交互,这样对象就具有较强的独立性、自治性和模块性。

2. 类

类是一组具有相同属性和相同操作的对象集合,类的属性用来表示对象的各种特征,类的操作是对于对象行为的描述,包括操作名称和操作实现过程。把众多的事物归纳成一些类是人们认识客观世界时的常用方法。分类所依据的原则是抽象,即忽略事物的非本质特征,只关注与当前目标有关的本质特征,从而找出事物的共性,把具有共性的事物划为一类,得出一个抽象的类的概念。类与对象的关系是抽象与具体的关系,类是多个对象的综合抽象,对象是类的个体实例。

3. 消息

对象与对象之间并不是彼此孤立的,它们之间存在联系,在面向对象的系统中,对象之间的联系是通过消息传递进行的,消息是对象之间相互请求和相互协作的途径,是要求某个对象执行其中某个功能操作的规格说明。通过发送消息操纵对象,对象接收消息,根据消息及消息参数调用相关的服务,进行处理并予以响应,从而实现系统功能。

3.2.4 基本特征

面向对象的方法认为客观世界是由各种“对象”所组成的,任何事物都是对象,每一个对象都有自己的运动规律和内部状态,每一个对象都属于某个“类”。复杂的对象可以是由相对简单的对象以某种方式而构成的。通过类比,发现对象间的相似性,即对象间的共同属性和行为,这就是构成类的依据,对象间的相互联系是通过传递“消息”来完成的,消息就是通知对象去完成一个允许作用于该对象的操作。

(1) 抽象性:关注与当前目标有关的本质特征,忽略非本质特征,找出事物的共性,归为一类,就会得到一个抽象的概念。抽象包括两个方面,一是过程抽象,二是数据抽象。过程抽象是指任何一个明确定义功能的操作都可被使用者看作单个的实体。数据抽象定义了数据类型和施加于该类型对象上的操作,并限定了对象的值只能通过这些操作来访问和修改。

(2) 封装性:它将属性和操作结合在一个类中,对象的属性一般不被外界直接访问,而是通过对象的操作来读取和修改;封装隐蔽对象的内部细节,只留少量接口,接收外界的消息。封装保证模块具有良好的独立性,便于系统维护,对系统的修改仅限于类的内部,是面向对象的特征之一。

(3) 继承性:广义地说,继承是指能够直接获得已有的性质和特征,而不必重复定义。在面向对象的软件技术中,继承是子类自动地共享父类中已定义的属性和方法。子类继承父类的所有属性和方法,避免了许多重复性的工作,因此子类的属性与操作有自己定义的,也有从父类继承来的。继承是传递的,当子类被更下层的子类继承时,它所继承的和自己定

义的属性和操作又被下一层继承下去。继承是面向对象方法中最显著的特点,提高了软件的可重用性,使得软件的可扩充性大大加强。类的继承关系如图 3-2 所示。

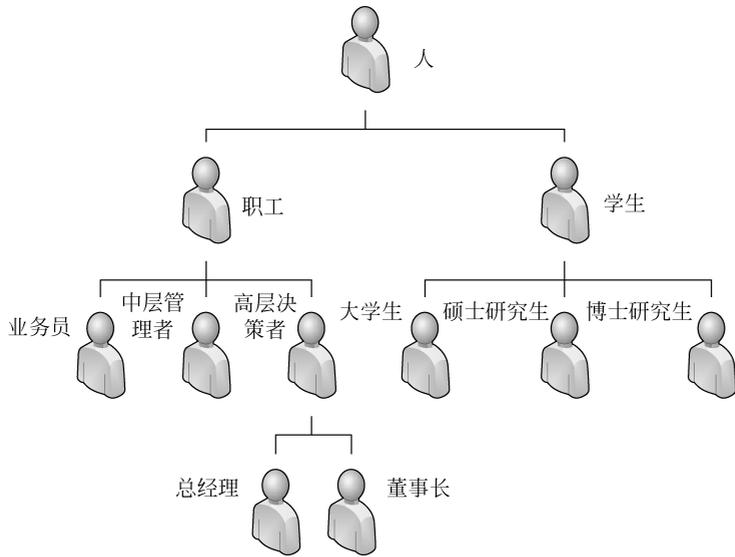


图 3-2 类的继承关系

(4) 多态性: 相同的方法作用于类型不同的对象上可以得到不一样的结果, 每个对象通过满足自身条件的方式响应同样的消息, 使得软件的可重用性和灵活性进一步增加。例如打开窗户的操作, 基于窗户的不同, 可以得到平开、推拉、上悬等多种不同的结果。

继承性和多态性的结合, 可以生成一系列相互类似但独一无二的对象。继承性使得这些对象共享许多相似的特征, 多态性针对相同的消息, 使得不同对象可以有独特的表现方式, 实现个性化的设计。

3.2.5 开发过程

按照系统开发生命周期的理论, 面向对象的信息系统开发方法可以分为以下 5 个阶段。

1. 面向对象的分析

面向对象分析(Object Oriented Analysis, OOA)是系统开发过程中的问题定义阶段, 目标是完成对所求解问题的分析, 确定系统“做什么”, 并建立系统模型。

面向对象的分析需要对问题域和系统责任进行分析和理解, 问题域有哪些值得考虑的事物, 面向对象的分析模型中就有哪些对象, 而且强调对象的命名与客观事物一致, 找出描述它们的类和对象, 定义其属性和操作。用一般-特殊结构描述事物间的继承关系, 用整体-部分结构描述事物间的组成关系, 用实例连接和消息连接表示事物间的静态联系和动态联系, 最终获得一个符合用户需求, 并能够反映问题域和系统责任的面向对象分析模型。

通过面向对象分析建立的系统模型是以对象概念为中心的, 它由一组相关的类组成。面向对象分析可以采用自顶向下的方法, 逐层分解建立系统模型, 也可以自底向上从已定义

的类出发,逐步构造新类。

2. 面向对象的设计

面向对象的分析建立了反映问题域的对象分析模型,不考虑与系统具体实现有关的因素(如编程语言、用户界面、数据库等),从而使面向对象的分析模型独立于具体实现。面向对象的设计(Object Oriented Design,OOD)则是面向对象系统开发方法在设计阶段应用与扩展的结果,是将面向对象分析阶段所创建的分析模型转换为设计模型,解决“如何做”的问题。面向对象设计的目标是产生一个满足用户需求、可实现的设计模型。

面向对象的设计内容主要包括两部分:一是不经过转换,仅做部分必要的修改和调整,把面向对象的分析模型直接用于面向对象的设计,作为面向对象设计的一个部分;二是针对具体实现中的人机界面、数据存储、任务管理等因素补充与实现有关的部分,采用面向对象的分析方式分析相同的表示法和模型结构。

3. 面向对象的编程

面向对象编程(Object Oriented Programming,OOP)的任务是将从面向对象分析和设计得到的模型用程序加以实现,即采用面向对象的程序设计语言,为面向对象的设计模型的每个成分编写程序。理想的面向对象开发规范,要求在面向对象的分析、设计阶段就对系统中的对象及其内部构成(属性和操作)与外部关系(静态和动态联系)有透彻的认识和清晰的描述,而不是把这些问题留给程序员去思考。程序员只需用具体的数据结构定义对象的属性,用具体的语句实现流程图所表示的算法。

4. 面向对象的测试

面向对象的测试(Object Oriented Testing,OOT)以类作为基本测试单位,差错范围主要是类定义之内的属性和操作,以及有限的对外接口(消息)所涉及的部分,可以大大减少错误的影响范围。一个类通常包括一组不同的操作,而一个操作也可能存在于一组不同的类中,所以类的测试不能再孤立地测试单个操作,而应该把操作作为类的一部分。此外,由于继承性的存在,面向对象的测试在完成对父类的测试后,子类的测试重点只是那些新定义的属性和操作。

5. 面向对象的维护

在面向对象的维护(Object Oriented System Maintenance,OOSM)中,程序与问题域是一致的,各个阶段的表示也是一致的,从而减少了理解的难度。无论是发现程序中的错误而逆向追溯到问题域,还是需求发生变化而从问题域正向跟踪到程序,都是很方便的。

3.2.6 特点

1. 面向对象开发方法的优点

(1) 良好的可复用性:基于类建立的系统模型,与基于“过程”和“数据”建立的系统模

型相比,更具稳定性,增强了系统的适应性,对复用支持程度高。

(2) 易于维护: 由于对象和类的规范性,维护人员易于理解运行过程和原理,可维护性好。

(3) 良好的可扩充性: 以对象和类为基础,实现了从对客观世界对象客体的描述到软件结构的直接转换,大大减少了后续软件开发量,缩短了开发周期。

2. 面向对象开发方法的缺点

(1) 面向对象方法的关键是从客观世界抽象出对象,但是客观世界的复杂性,使得完成对象的抽象比较困难。

(2) 面向对象的开发方法,需要有一定的软件基础支持才能应用。

(3) 如果大型系统开发中,一开始就采用自底向上的面向对象方法开发系统,而不经自顶向下的整体划分,易造成系统结构不合理、各部分关系失调等问题。因此,面向对象的开发方法与结构化系统开发方法在系统开发中相互依存、不可替代。

3.3 计算机辅助软件工程

3.3.1 基本思想

计算机辅助软件工程(Computer Aided Software Engineering, CASE)是计算机技术在系统开发活动、技术和方法中的应用,是软件工具与开发方法的结合体,它使得人们能在计算机的辅助下进行软件开发,为计算机软件开发的工程化、自动化进而智能化打下基础。计算机辅助软件工程是提高系统开发效率与质量的重要途径。如果严格地从认知方法论的角度来看,CASE 是技术,但从 CASE 的发展对系统开发过程所支持的程度来看,又不失为一种实用的系统开发方法。

具体地说,CASE 能生成各种需求分析、功能分析和结构图表(如数据流图、结构图、实体联系图等),进而成为支持整个系统开发全过程的一种大型综合系统,能支持除系统调查之外的所有系统开发过程,它可以按照系统开发商规定的应用规则,由计算机自动生成合适的计算机程序。CASE 的实质是为系统开发人员提供了一组优化的、集成的、且能节省人力的系统开发工具,实现软件生命期各阶段的自动化,帮助开发者方便、快捷地产生出系统开发过程中的各类图表、程序和说明性文档,着眼于系统分析、设计、开发、实施和维护等各个环节的自动化,使开发工作成为以自动化工具和支撑环境支持的自动化过程,并使其成为一个整体。

CASE 的目的是使开发支持工具与开发方法结合起来,通过实现分析、设计与程序开发、维护的自动化,提高信息系统开发的效率和信息系统的质量,最终实现系统开发的自动化。

3.3.2 基本功能

1. 中心信息库

中心信息库是存储和组织所有与应用软件系统有关的信息机构,提供人机界面以进行有效的管理,包括系统的规划、分析、设计、实现和计划管理等信息,如结构化图形、屏幕与菜单的定义、报告的模式、记录说明、处理逻辑、数据模型、组织模型、处理模型以及源代码等。中心信息库具有对系统信息存储、访问更新、分析和生成报告的功能,系统开发人员可以直接从中获取所需的信息,合理地组织与管理系统开发的工作过程。

2. 图形功能

图形实际上是软件模型化的语言,它为软件的描述提供了一种简明的、重构的、没有歧义的方法,快捷地产生系统开发过程中的各类图表、程序和说明性文档,是产生好的系统和程序文档的基础。清晰的图形在复杂系统的开发和编程的过程中起着关键性的作用,它能为开发人员提供清楚的思路,提高工作速度并改进产品的质量,更是一种重要的沟通工具。

3. 查错功能

在系统开发中,尽早排除错误是降低成本的一种行之有效的方法,CASE 提供了以系统说明书为依据的自动检查的功能。

4. 支持系统的原型设计

CASE 为建立原型提供了各种工具,如屏幕绘图程序、报告检测,达到系统的一致性和完整性。借助于 CASE 工具,系统开发人员可对原型进行模拟运行以验证系统设计模型的正确性。

5. 代码自动生成

CASE 通过程序设计规格说明书,使用户在较短时间内,自动或半自动地生成所需的代码段落,实现编程阶段的自动化。这种自动生成可能是一个框架,也可能是一个完整的程序,极大地提高了系统开发的效率。

6. 有利于应用结构化方法

CASE 提供的若干工具有利于结构化分析、结构化设计和结构化编程,从而使结构化方法实现自动化。CASE 工具为画数据流图、E-R 图等结构化图提供了图形支持,同时可自动生成诸如系统说明和伪代码等形式的规格说明。同时,CASE 指导用户正确地使用结构化方法,要求用户按照一定的标准化次序进行系统分析与设计。

由于系统开发涉及复杂的技术背景和管理环境,人在系统开发各阶段中始终处于关键地位。全部开发工作的自动化是不切实际的幻想,但采用 CASE 方法可辅助人们更快、更好、更低成本地进行系统开发。特别是基于人工智能的 CASE 方法,它将对系统开发产生

重大的影响。

3.3.3 特点

(1) 计算机辅助软件工程提供了一组能够自动覆盖软件开发生命周期各个阶段(除系统调查外)的集成工具,强有力地支持了系统开发的全过程,解决了从客观对象到软件系统的直接映射问题。

(2) 支持自顶向下的结构化系统开发方法,使结构化系统开发方法更加实用;支持自底向上的原型化和面向对象的方法,使原型化和面向对象的方法能很好地付诸于实施。

(3) 实现分析设计图表和程序编写自动化,使开发者从繁杂的分析设计图表和程序编写工作中解放出来,自动生成统一的、标准化的系统开发文档,减轻了相关人员的工作负担,加快了软件开发速度。

(4) 项目管理、分析、设计、编程、测试和维护工具辅助系统的开发,加速了系统的开发过程,简化了软件的管理程序,提高了软件的重用性,而且自动检测的方法大大提高了软件的质量,提高了系统的维护能力。

本章小结

信息系统开发方法提供了对完成系统生命周期的每一阶段的详细指导,本章主要介绍了结构化系统开发方法、面向对象系统开发方法和计算机辅助软件工程方法,它们各自遵循着一定的基本思想,适用于一定的范围,解决问题的出发点和侧重点也各不相同,但共同的目标是提高系统开发的效率和系统的质量。

结构化系统开发方法是用系统工程的思想 and 工程化的方法,按照用户至上的原则,基于系统开发生命周期,严格划分系统阶段,结构化、模块化、自顶向下地分析和设计系统。适合需求明确的信息系统开发,但开发周期长、灵活性差。

面向对象系统开发方法是基于面向对象的程序设计方法逐步发展的,它把数据和过程包装成为对象,以对象为基础对信息系统进行处理,是一种综合性的开发方法,具有良好的可复用性和可扩充性,易于维护。但初学者对相关概念不易理解,需要有一定的软件基础,整体划分不当易造成系统结构不合理、各部分关系失调等问题。

计算机辅助软件工程的主要目标是使用自动化工具自动生成信息系统,本章介绍了该工具应该具备的基本部件和计算机辅助软件工程方法的特点,但目前该方法只能部分支持信息系统的开发过程。

本章练习

1. 问题思考

- (1) 结构化开发方法的基本思想是什么? 它的缺点是什么?
- (2) 结构化开发方法将系统开发分为几个阶段? 各阶段的任务是什么?

- (3) 面向对象系统开发方法的基本思想是什么？
- (4) 面向对象系统开发方法和结构化开发方法的区别是什么？
- (5) 计算机辅助软件工程的含义是什么？
- (6) 面向对象开发方法的开发过程分为哪些阶段？

2. 专题讨论

- (1) 试分析下面公式的含义。

面向对象 = 对象 + 类 + 类继承性

- (2) 结构化开发方法认为：系统开发是一个在不同层次上抽象的过程。试分析其含义。