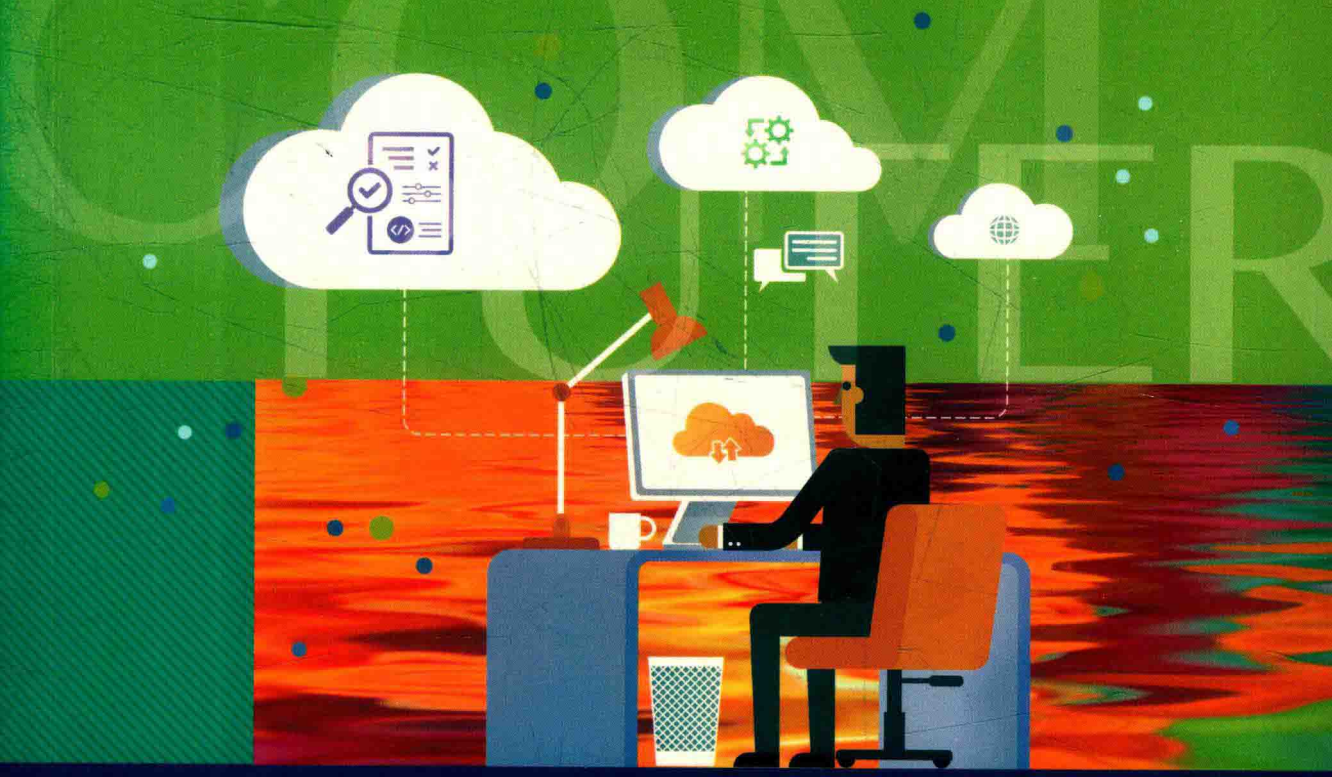
高等院校计算机系列规划教材



软件系统分析与设计任务驱动案例教程

◎ 苏春燕　主　编
◎ 邓　蓓　韩　红　孙　锋　副主编

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高等院校计算机系列规划教材

软件系统分析与设计任务驱动案例教程

苏春燕 主 编

邓 蓓 韩 红 孙 锋 副主编

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书介绍了软件系统典型的开发路线及其开发方法，且重点讲解了面向对象的软件系统开发的分析与设计方法，既包括理论知识、建模技术，又包括一些建模工具软件的使用技能。其内容安排是以一个面向对象的软件系统开发案例的分析与设计过程贯穿来讲解理论知识和设置实训任务。另外，书中关键术语和一些图例采用中英文两种表达方式，有利于读者掌握专业知识的同时掌握专业英语。

本书内容设置系统、连贯，叙述清晰，逻辑严密，且结合待开发案例讲述，使各知识点更易于理解。涉及工具软件使用的实训任务指导叙述准确、翔实，包括了多种建模软件的操作指导，易于学生掌握，且习题丰富。

这是一本适合应用型本科和高职高专的软件及信息管理类专业学生的教材，同时也是一本软件从业人员系统学习面向对象的软件系统分析与设计技术的入门书，当然它也包括较深入的知识。本书还适合作为有双语教学要求的此类课程的教材。

图书在版编目（CIP）数据

软件系统分析与设计任务驱动案例教程 / 苏春燕主编. —北京：电子工业出版社，2018.9
ISBN 978-7-121-34659-0

Ⅰ. ①软… Ⅱ. ①苏… Ⅲ. ①软件工程－系统分析－高等学校－教材②软件设计－高等学校－教材
Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字（2018）第 141754 号

策划编辑：左　雅
责任编辑：左　雅　　特约编辑：俞凌娣　朱英兰
印　　刷：北京虎彩文化传播有限公司
装　　订：北京虎彩文化传播有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787×1 092　1/16　印张：14.75　字数：424.8 千字
版　　次：2018 年 9 月第 1 版
印　　次：2018 年 9 月第 1 次印刷
定　　价：39.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254580　zuoya@phei.com.cn。

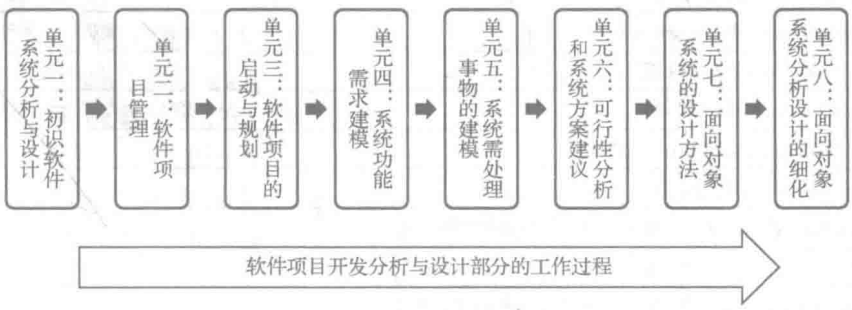
前　言

目前开发一个面向对象软件系统大致要做的工作有：①准确获取、记录和分析用户的需求；②考虑系统应包括哪些类的对象以及这些类的对象应如何相互协作才能实现这些需求；③用具体的编程语言来编写程序定义类、创建对象以实现用户对系统的需求。

掌握具体的程序开发语言只能解决第③步的问题，还不能高效完成一个满足用户需求的软件系统的开发。本书就是针对前面两步编写的，即如何对系统用户的业务需求进行“获取、记录和分析”，又如何从技术和实现的角度来“设计”一个软件系统以满足这些业务需求，同时如何用模型来记录设计方案。

本书介绍了软件系统典型的开发路线及其开发方法，且重点讲解了面向对象的软件系统开发的分析与设计方法，既包括理论知识、建模技术，又包括一些建模工具软件使用的技能，是理论和实际密切结合的一本教材，且实训的任务指导部分准确、翔实，并配有丰富的习题。书中涉及多种建模软件的使用，其中的 UML 建模软件主要讲解了经典的建模软件 Rational Rose 的使用，但同时给出了此类的开源建模软件 StarUML 和 JUDE-Community 的使用指导，引导学生课后用这些建模工具软件创建 UML 模型。还根据软件系统分析与设计中数据建模及交流项目任务、资源和时间安排的需要，讲解了用 Microsoft Visio 创建 E-R 图和用 Microsoft Project 创建项目进度表的方法，这些安排有利于提高学习者的实操能力，从而提高职业素养。本书适合作为应用型本科和高职高专学生的计算机软件及信息管理类专业学生的教材，同时也是一本软件从业人员系统学习面向对象的软件系统分析与设计技术的入门书，当然它也包括较深入的知识。

本书的一个重要特点就是“以软件系统开发的工作任务为导向划分教学单元”，具体参见下图。其中需要说明的是，单元一用简单实例展示面向对象软件系统开发分析与设计的主要过程、相关概念，引导学生初步了解、掌握常用的建模工具软件的使用方法，是开始分析与设计的基础；单元二是软件项目管理的基础知识和技能，其中的部分内容可以选学，是完成单元三的软件项目开发工作初始任务“启动与规划”的基础；单元八是对面向对象软件开发分析与设计的基本知识和技能的拓展，可以看作是知识和技能的螺旋式增长。



以工作任务为导向的教学单元的设置

本书以一个面向对象的软件系统开发案例的分析与设计过程贯穿始终讲解理论知识和设置实训任务，其间穿插设置一些所涉及的建模工具软件使用的实训任务以及一些拓展的知识和技能，引入了任务驱动及案例的教学设计，是探索、促进这类教学方式的一次努力和尝试。

另外，书中的关键术语和一些图例采用中英文两种表达方式，这样有利于读者掌握专业知识的同时掌握专业英语，因此还适合作为有双语教学要求的此类课程的教材。

关于全书的授课学时安排，对于高职高专学生而言建议 68 学时左右，其中理论和实训各占一半学时；对于应用型本科学生，这个学时可以酌情减少。教师可以根据学时重点讲授一些核心单元，其他单元选学或课后自学。对于高职高专学生的软件系统分析与设计类的课程建议至少 48 学时，理论和实训各占 24 学时，其中至少有 16 学时上机用工具软件绘制模型图。如果安排这些学时，单元二的“软件项目管理”和单元六的“可行性分析和系统方案建议”可以安排课后学生自学或选学；对于应用型本科的此类课程，上机学时可以酌情减少，一些用工具软件绘制模型图的任务安排让学生课后按照任务指导的提示自学完成。最后要说明的是，附录 A 给出了待开发软件系统的案例背景资料和该系统的面向对象分析与设计的建模要求，如果课时允许可以安排学生完成，以检验和巩固前面所讲的知识和技能。

本书由苏春燕担任主编并负责统稿，由邓蓓、韩红和孙锋担任副主编。本书参考了多本国内外同类教材，借鉴了我校（天津中德应用技术大学）与加拿大不列颠哥伦比亚理工大学（BCIT）计算机学院多年合作办学的经验，也包含了我校的“软件项目开发方法”市级精品课、“软件系统分析与设计”校级精品资源共享课和“面向对象软件系统分析与设计”校级优质课建设的成果，在此对课程建设团队所有成员的付出表示诚挚的谢意！

虽然在编写本书过程中力求完美，但由于经验不足等原因，难免有疏漏之处，敬请各位读者批评指正。

为便于教学，本书提供书中内容的电子课件、习题参考答案，以及实训中涉及的工具软件建模方法的 15 段视频，具体视频内容请见下表，希望给大家的教学与学习带来方便。

序　号	视 频 内 容	对 应 实 训
1	用 Rational Rose 实现正向工程	实训一
2	用 Microsoft Project 开发项目基本进度表	实训二
3	用 Microsoft Visio 绘制系统关联图	实训三
4	用 Microsoft Project 开发带资源分配的项目进度表	实训四
5	用 Rational Rose 绘制用例图	实训五
6	用 Rational Rose 绘制活动图	实训六
7	用 Rational Rose 绘制系统顺序图	实训六；实训十二（可参考借鉴）
8	用 Microsoft Visio 绘制 E-R 图	实训七
9	用 Rational-Rose 绘制分析类图	实训八；实训十二（可参考借鉴）
10	用 Microsoft Excel 计算项目的净现值	实训九
11	用 Rational-Rose 绘制通信图	实训十
12	Rational Rose 通信图消息位置的改变与转换为顺序图方式	实训十
13	用 Rational Rose 实现逆向工程	实训十一
14	用 Rational Rose 创建具有方法参数及继承关系的设计类图	实训十一；实训十二（可参考借鉴）
15	用 Rational Rose 绘制状态机图	实训十四

注：实训十五和实训十六的建模可参考借鉴序号 5、6、7、8、9、12 和 14 的视频。

编　者

目　录

单元一　初识软件系统分析与设计

从前言中我们已经了解到本书关注的是如何获取和分析用户需求，又如何综合利用技术设计一个软件系统来满足用户的需求，以及系统设计方案的记录方式。在开始学习之前需要对系统分析与设计的概念和相关知识、技能有个初步、概括的了解，由于软件开发目前主要采用面向对象的编程语言，因此，我们在这个单元主要介绍软件系统分析与设计的基础知识和技能。

任务 1.1　认识软件系统分析与设计

内容引入

在初步了解“软件系统分析与设计”时，人们最关心哪些内容呢？有可能是下面几点：

- ✓ 软件系统的开发大致要经历哪些阶段？各个阶段主要达成什么目标？
- ✓ 软件开发涉及哪些人员？
- ✓ 什么是软件系统的分析与设计？
- ✓ 目前典型的面向对象软件系统开发的分析与设计过程要建立哪些模型？

本任务将认识这些与软件系统开发相关的概要知识，下一任务将初识面向对象软件系统分析与设计的建模工具软件 Rational Rose 和 StarUML 等的使用方式。

学习目标

- ✓ 理解软件系统开发的大致过程、驱动力和关联人员。
- ✓ 理解系统开发生命周期的类型划分。
- ✓ 理解系统开发方法、工具、模型和技术的区别与联系。
- ✓ 理解面向对象软件系统的开发过程、涉及的技术，理解目前较流行的开发方法——RUP 的开发阶段的划分方式。

1.1.1　软件系统开发的上下文

图 1-1 展示了软件系统开发的相关因素，其左面是系统开发的“参与者”，也称“系统关联人员”。右面是系统开发涉及的主要“过程”，不同的参与者主要参与的过程不同，系统视图也不同。上面是促使系统开发的“业务驱动力”，如：经济全球化环境下，需要开发的支持贸易的软件系统需要支持多种语言、货币汇率、国际贸易规则和不同商业文化的业务方式等；电子商务环境下，要求企业处理日常事务的软件是基于 Web 体系结构的；业务驱动力还包括企业业务过程持续改进和全面质量管理对软性系统的开发要求。下面是促使系统开发的“技术驱动力”，如：网络技术、移动无线技术的发展，企业资源规划（ERP）、供应链管理（SCM）、客户关系管理（CRM）和企业应用集成（EAI）等可购买的企业应用软件的出现。图 1-1 所涉及的与系统分析设计相关的概念描述如下。

系统分析（System Analysis）：理解并详细说明一个软件系统应该做什么的过程。

系统设计（System Design）：详细说明系统的许多组件在物理上怎样实施（分析阶段确定的）系

统功能的过程。

系统分析员（System Analyst）：利用分析技术确定用户的业务需求，并利用信息技术设计出满足业务需求的系统实施总体方案建议。他们可以看成是业务人员和技术人员之间的桥梁。

另外，小型软件开发项目中可能分析、设计、编程职责集中于一、两个人，就需要一般软件开发人员具备这些综合的知识、技能。因此，系统分析与设计技术也是一般软件编程人员需要掌握的知识。

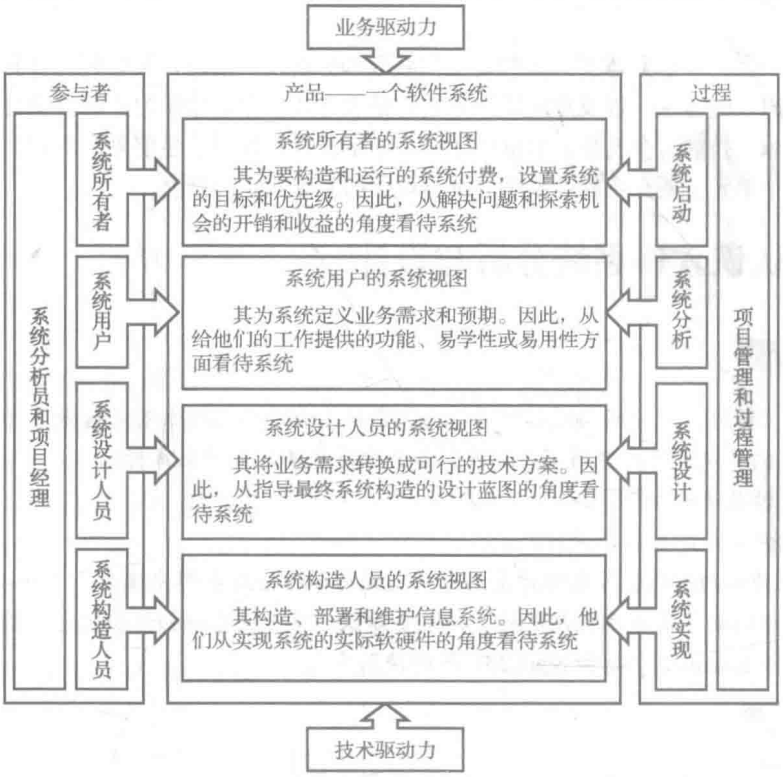


图 1-1　软件系统开发上下文

1.1.2　软件系统开发生命周期概念与类型划分

1. 系统开发生命周期

系统开发生命周期（the System Development Life Cycle，SDLC）：建立、部署、使用和更新一个软件系统的整个过程。

可根据方法是否具有预测性和适应性对系统开发生命周期路线进行分类，从完全预测到完全适应，允许将这两个分类点连成一个连续体，如图 1-2 所示。

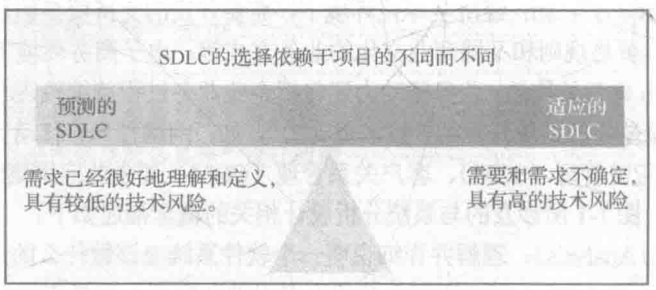


图 1-2（a）　系统开发生命周期的预测-适应方法（中）

（1）预测方法（Predictive Approach）：一种可以预先规划并组织的软件项目开发，即要求根据规划对新软件系统进行开发的系统开发生命周期方法。

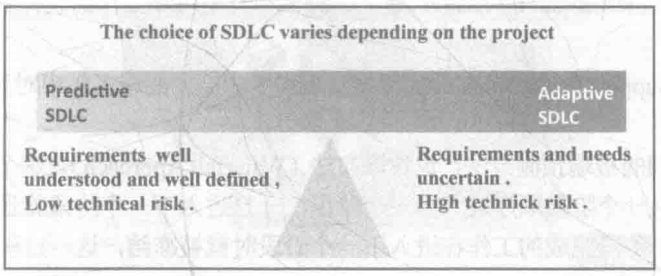


图 1-2（b）　系统开发生命周期的预测-适应方法（英）

例如，一个系统开发项目的需求是“将一个老的主机库存系统转化为一个基于网络的客户/服务器架构的系统”，这个待开发的软件项目的需求非常明确、清晰，而使用的客户/服务器架构的技术也比较成熟，因此这个项目就可以事先详细规划进度和设计说明，然后据此构造项目。那么这个项目就采用了系统开发生命周期的预测方法。

（2）适应方法（Adaptive Approach）：一种不能预先规划的软件开发，即要求在开发进展过程中进行调整的灵活的系统生命周期方法。

这种方法用于系统和用户的需求不明确的时候，这种情况下项目不能预先规划，在初步开发工作后还需要确定系统的一些需求。开发人员制定的解决方法就必须灵活，在项目开发进展中可以调整。

大部分软件系统开发的项目既有预测的元素，又有适应的元素，而不是仅位于天平的两端。预测方法发明于 20 世纪 70～90 年代之间，更为传统，那时系统开发采用结构化程序设计语言，其修改和扩展都较为复杂和费时。而适应方法是随着面向对象方法发展起来的，始于 20 世纪 90 年代并一直发展至今，这一时期系统开发多采用面向对象的程序设计语言，其修改和扩展就相对容易，也为在系统开发过程中灵活调整项目需求提供了技术基础。

2. 传统系统开发生命周期预测方法

下面对于传统系统开发生命周期预测方法（Traditional Predictive Approach to the SDLC）的系统开发阶段（System Development Phases）进行描述。图 1-3 显示了传统系统开发生命周期预测方法的五个阶段及其之间的常见关系。

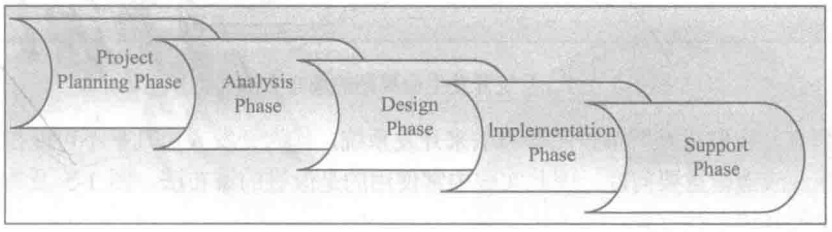


图 1-3　系统开发阶段及其之间常见关系

（1）计划阶段（Planning Phase）：确定新系统的作用域、确定项目的可行性、制订进度表和资源分配计划并进行项目其余部分的预算。

（2）分析阶段（Analysis Phase）：理解、定义新系统的业务需求及其优先级，并确定系统实施的总体方案建议。其主要作用是获取系统是做什么的。

（3）设计阶段（Design Phase）：根据分析阶段确定的业务需求、其优先级和实施的总体方案建议，设计出系统物理解决方案。其主要作用是决定系统如何做。

（4）实施阶段（Implementation Phase）：建立、测试和安装可靠工作的软件系统，培训用户并使其受益于系统的使用。

（5）支持阶段（Support Phase）：在系统初始安装和生命周期的许多年中对系统进行升级和维护，以保持系统有效运行。

系统开发生命周期的极端预测方法，也称瀑布法（Waterfall Approach），这个名字形象地描述了其极端的实现方式，即从一个阶段顺序进入另一个阶段，一旦进入下一个阶段就像瀑布一样不能回到上一个阶段，这样上一个阶段完成的工作在进入下一个阶段时就被冻结，这一过程说明参见图 1-4。

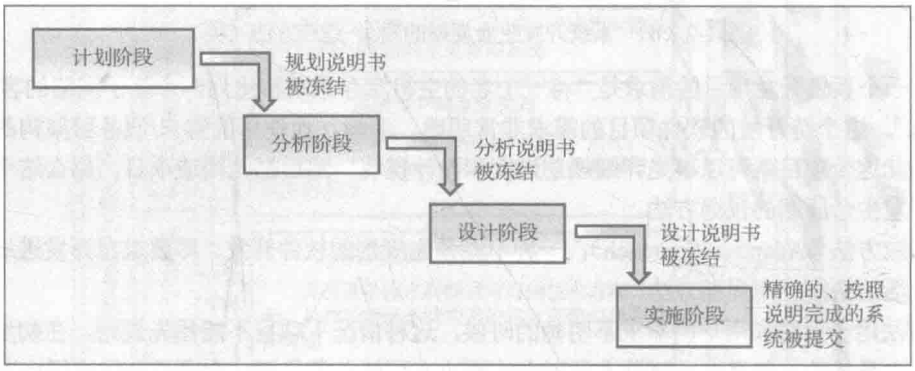


图 1-4（a） 系统开发生命周期的瀑布方法（中）

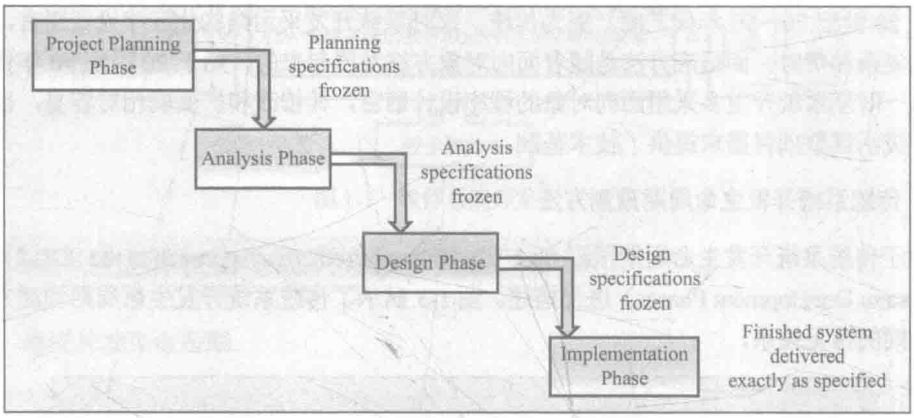


图 1-4（b） 系统开发生命周期的瀑布方法（英）

大家不难想到，我们很难严格按照瀑布法来开发系统，作为开发人员几乎不可能在完成一个阶段时不出现任何错误或遗留重要问题，因此实践中常使用的是改进的瀑布法。图 1-5 及其文字描述说明了改进瀑布方法的具体实践方式。

下面是关于图 1-5 的解释说明。

- ✓ 仍需要开发一个十分透彻的项目规划，但其他每个阶段相互重叠、影响和依赖。
- ✓ 在开始设计之前做一些分析，但在设计中会发现在需求方面需要更多的细节，或一些需求是原先没有想到的，因此要返回去附加一些分析。
- ✓ 出于效率原因，在分析需求时，可能考虑并设计各种报表或表格。
- ✓ 为帮助理解用户的需求，可能要提早实现最终系统的一部分。

✓ 各个阶段之间不能完全重叠，其部分原因是：相关依赖性。

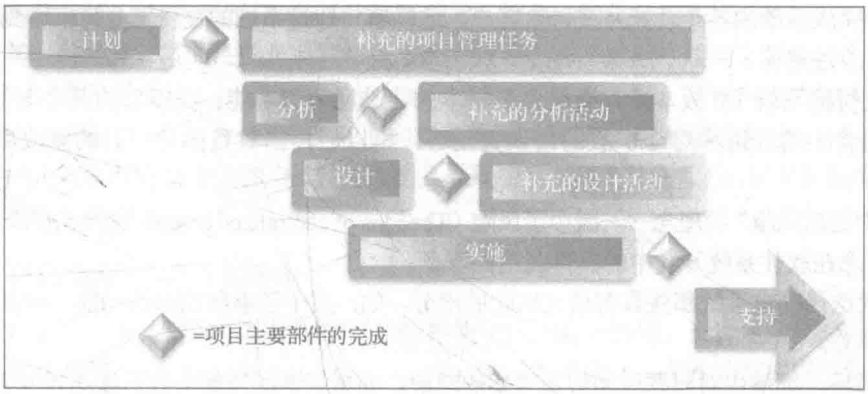


图 1-5（a）　系统开发阶段的重叠（中）

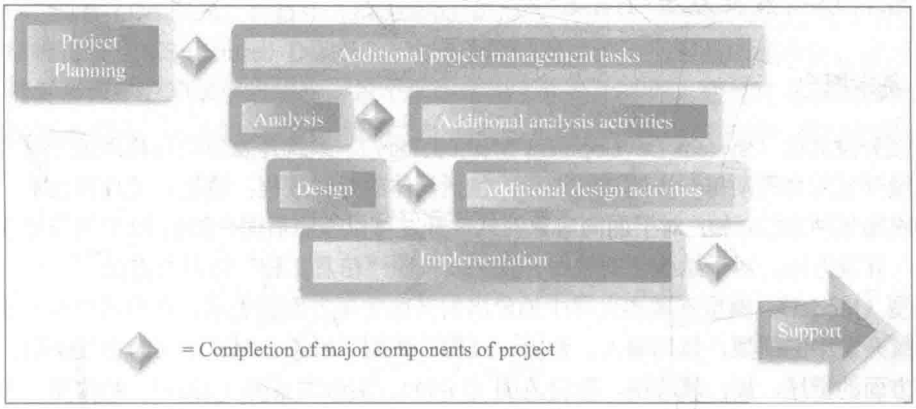


图 1-5（b）　系统开发阶段的重叠（英）

3. 新的系统开发生命周期适应方法

这种在系统开发过程中可动态改变计划和模型的 SDLC 适应方法的特点总结如下。

（1）螺旋式开发（Spiral Development）。其特点是：项目循环顺序经过所有的开发活动，即计划、分析、设计、构造、测试，反复进行这些活动直到项目完成；每次循环都会产生一个原型（Prototype）。螺旋式开发的生命周期模型如图 1-6 所示。

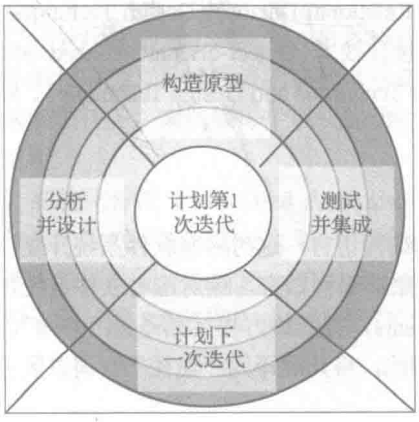


图 1-6（a）　螺旋式开发的生命周期模型（中）

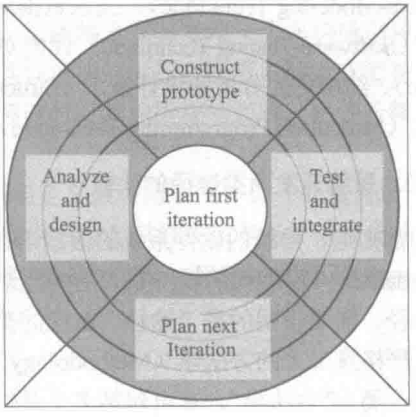


图 1-6（b）　螺旋式开发的生命周期模型（英）

（2）增量式/进化式开发（Incremental/Evolutionary Development）。增量式/进化式开发的特点就是逐步、分阶段完成系统的各个功能并交付给客户，这样客户通过系统的交付部分可以看到系统的开发进度，并提出改进建议。

如：在交付的系统 1.0 版本中，将包含最基本的、最重要的功能；在以后的某个时间，交付 1.1 版本，其中包含小的改进和对 1.0 版的错误更正等；再以后，当对整体进行大的修改后，交付 2.0 版本。

（3）“集中处理风险”的理念——减少了风险（The idea of centralized processing the risks——to reduce risk）。这一理念在软件系统开发中的具体应用如下。

- ✓ 第一次迭代确定系统那些具有最大风险的部分，如：某个子系统或系统功能；涉及新技术的可行性部分。
- ✓ 每次循环，都增进对问题域和解决方案的理解，并根据新的理解进行后续迭代开发。

1.1.3 软件系统开发方法与途径

1. 基本概念

（1）系统开发方法（System Development Methodology）。系统开发方法也称系统开发方法学，提供了完成系统开发生命周期每一步的详细指导，包括具体的开发过程、模型、工具和技术。一般企业开发软件都遵循某种方法，如：对于面向对象的软件开发可以按照后面介绍的 RUP 开发方法，也可以按照“敏捷”开发方法；对于结构化的软件开发可以按照“信息工程”等开发方法。

（2）模型（Model）。模型是现实世界中抽象出的某些重要方面的表示，其形式通常为图和表。其中一类是系统开发中的模型，包括输入、输出、过程、数据、对象、对象之间的相互作用、位置、网络和设备等方面的描述，如：流程图、数据流图（DFD）、实体关系图（ERD）、结构图、用例图、类图和顺序图等；另一类模型是项目规划模型，如：PERT 图、Gannt 图和组织层次图等。

（3）工具（Tool）。工具是指帮助生成项目中所需模型或其他组件的软件支持，比如：集成开发环境（IDE）、项目管理软件（Project Management Application）、制图软件（Drawing/Graphics Application）、计算机辅助系统工程（CASE）工具（Computer-Aided System Engineering Tool）、数据库管理软件（Database Management Application）、反向工程工具（Reverse-Engineering Tool）软件、代码生成工具（Code Generator Tool）软件即正向工程软件。

（4）技术（Technology）。技术指帮助分析员完成系统开发活动或任务的一组方法，如：数据建模技术（Data-Modeling Technique）、关系数据库设计技术（Ralational Database Design Technique）、软件测试技术（Software-Testing Technique）、面向对象的分析和设计技术（Object-oriented Analysis and Design Technique）、结构化的分析和设计技术（Structured Analysis Technique and Design Technique）、结构化的编程技术（Structured Programming Technique）。

2. 系统开发两类途径的比较

我们都知道，传统的软件系统的编程语言是结构化的编程语言，这对应着软件系统开发的传统途径（Approache）；而现代的软件系统的编程语言是面向对象的编程语言，这对应着软件系统开发的面向对象途径。需要说明的是“途径”对应的英文是 Approache，有些地方将其翻译为“方法”，我们是为了与更严格意义上的方法学（Methodology）相区别，因此，将其翻译为“途径”一词以区分。它可以理解为一类 “方法学”，其可包括多个具体“方法学”。

（1）传统开发途径（Traditional Development Approach）。传统开发途径也称为结构化系统开发途径，包括结构化分析、结构化设计和结构化编程技术，简称结构化分析和设计技术（SADT）。信息工程（IE）是其一种形式，其关注的是系统需处理的“过程”和“数据”这两个部分。

（2）面向对象开发途径（Object-Oriented Development Approach）。面向对象开发途径包括面向对象分析（OOA）、面向对象设计（OOD）和面向对象编程（OOP），它把信息系统看作是一起工作来完成某项任务的相互作用的对象的集合，其关注的是系统应具有的“对象”，“对象”是系统需处理的“过程”和“数据”的集合体。

3. 传统结构化程序开发简介

这里逆向说明一下传统的结构化程序开发的相关知识，即从编程、设计逆向到对分析进行说明。

（1）结构化编程（Structued Programming）。这一阶段的要求是：提高计算机程序的质量；编写其他程序员容易理解和修改的代码；每个程序模块只有一个开始和结束；程序由三种结构组成，即顺序（Sequence）、判断（Decision）和循环（Repetition）。

（2）结构化设计（Structured Design）。这一阶段的基本设计思想是：模块化、自顶向下的程序设计，即把复杂程序分解成模块层次，每个模块实现一个基本功能，通过在高层的模块调用、执行较低层的模块来实现一些功能。模块和模块之间的关系可以用一种叫结构图的模型来图形化地表示，如图 1-7 所示。

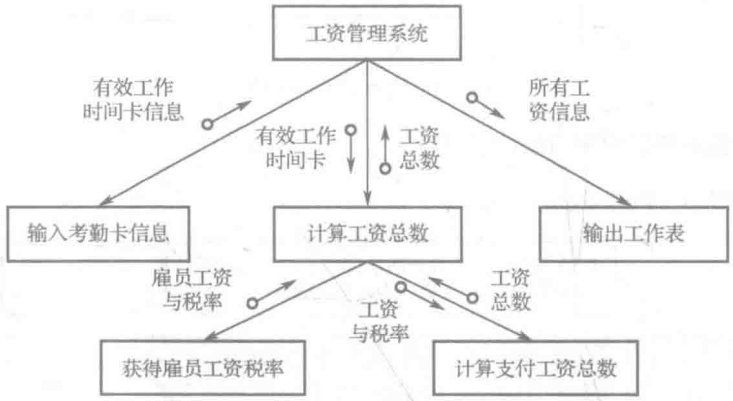


图 1-7　结构图形式表示模块间的关系

结构图（Structure Chart）是用结构化设计技术生成的显示程序模块层次的图示模型。

程序设计模块的两个原则是：松散耦合和高度内聚。松散耦合意味着每一个模块应尽可能地与其他模块保持相对独立，这使该模块在以后修改时不干扰其他模块的运行；高度内聚意味着每个模块实现一个清晰的任务，来明确每个模块的功能，避免以后修改时对其他模块的影响，也便于模块在其他程序的复用。

（3）结构化分析（Structured Analysis）。这一阶段需要定义系统需要做什么（处理需求）、系统需存储和处理的数据（数据需求）、输入和输出，以及它们如何在一起共同完成任务。结构化分析的结果一般用逻辑数据流图（Data Flow Diagram，DFD）和实体关联图（Entity Relationship Diagram，ERD）来显示和记录。图 1-8 和图 1-9 分别是它们的示例。

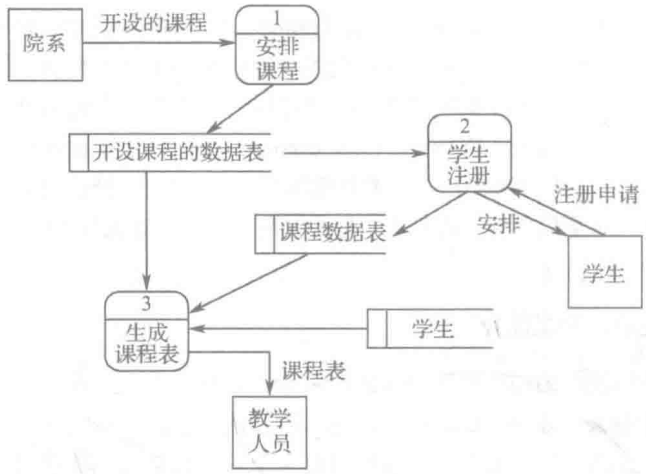


图 1-8　逻辑数据流图示例

图 1-9　实体关联图示例

4. 面向对象系统的分析与设计举例

（1）定义业务用例（Define Use Cases）。

目前对于用例可以从下面两点理解，后面将给出这一概念的更系统的定义。

✓ 用例表达的是使用应用软件某个功能的场景、步骤。

✓ 定义业务用例就是进行系统需求分析。

（2）定义领域模型（Define Domain Model）。

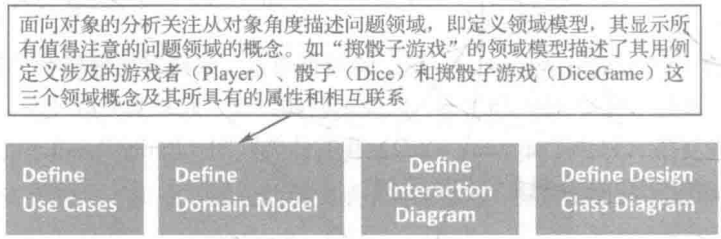


图 1-10 是根据前面“掷骰子游戏”的用例定义所确定的领域模型。