

第3章



需求分析



本章引言

继软件开发的前期准备阶段和软件可行性分析阶段之后,下一个非常关键的阶段就是需求分析阶段。需求分析就是分析软件用户的需求是什么。这一阶段之所以重要,就因为其具有指导性、决策性和方向性,在软件开发流程中具有举足轻重的作用。我们要对需求分析阶段产生足够的重视,因为在一个正规的应用系统开发过程中,需求分析的作用要远远大于程序设计本身的作用。



教学目标

1. 了解需求分析的目的和任务。
2. 掌握需求分析的过程。
3. 学会数据流程图的设计、数据字典的编写。
4. 学会软件需求说明文档的制定。



基础理论和实例

3.1 需求分析的任务

开发软件系统最为困难的部分就是准确地说明开发什么,难以处理的概念性工作就是编写出详细的技术需求,其中包括所有面向用户、面向机器和其他软件系统的接口。同时这部分一旦做错,最终将会给系统带来极大的损害,并且以后对其进行修改也极为困难。

在软件生命周期中,需求分析是用来说明开发什么的阶段,是一个极其重要的阶段。需求分析的根本任务就是为了满足用户的需求而确定系统必须实现的功能。所谓需求,应该是来源于用户调查,来源于某个行业的某些抽象模型的提炼,并参照行业规范进行业务分析的结果。值得注意的是需求是随时变化的。

3.1.1 需求分析的基本概念

需求分析就是分析软件用户的需求“是什么”,回答所要开发的应用系统将要“做什么”。通过对所要开发的目标系统的功能和性能进行详细的分析,用科学的方法来表述所要开发系统的逻辑方

案，建立系统的逻辑模型，从而设计出一个合理的优化系统，确定系统的开发方向。

目标系统模型的建立过程如图 3-1 所示，虚线部分是需求分析阶段所要解决的问题。需求分析所要做的工作是深入描述软件的功能和性能，确定软件设计的限制和软件同其他系统元素的接口细节，定义软件的其他有效性需求。

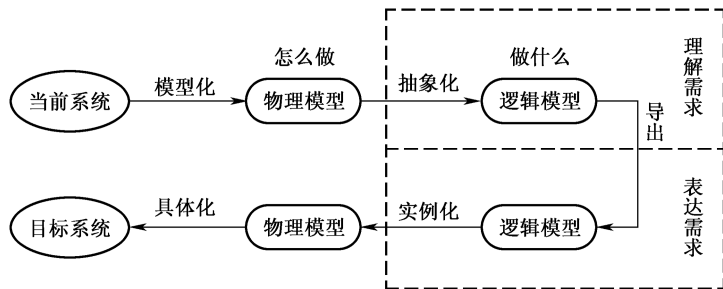


图 3-1 目标系统模型的建立过程

图 3-1 中，“理解需求”指的是软件开发人员对所开发的系统需求的理解。这里的“需求”包含以下三层意思：

（1）“需求”既包括了用户明确表达出来的要求，也包括了用户没有明确表达出来的隐含要求。需求分析员既要能够挖掘出用户没有明确表达出来的意思，也要能够通过需求分析修正和完善用户提出的要求。

（2）由于并不是用户的所有要求都是合理的，所以需求分析员必须全面理解用户的各项要求，但同时又不能一味地接受用户所有的要求。

对于其中模糊的要求要明确，然后再决定能否采纳。对于一些无法实现的要求，应该断然去掉并同时向用户做充分的解释，以求得理解。

（3）当用户和需求分析员在沟通上出现问题时，需求分析员要耐心地与用户交流想法，不断探讨需求内容并做出适当修改，直到用户满意为止。

在设计新系统逻辑模型时，软件需求分析员和用户出现沟通障碍是正常的情况，这是因为：

- 从用户角度，由于每个人的经历不同、教育程度不同、对客观事物的看法和描述也不尽相同。往往有些用户对自己的工作业务非常熟悉，但要清楚地表达出来却比较困难。还有一些用户缺乏计算机背景知识，因而所提出的一些需求令系统分析员难以正确理解或实现。
- 从需求分析人员角度，虽然他们是软件开发方面的专家，却不一定擅长开发项目所涉及的领域，缺乏专业知识，从而使得用户和需求分析员之间由于缺乏共同语言而无法进行良好的沟通。

图 3-1 中，“表达需求”是需求分析员把所接受的用户要求通过逻辑模型准确地表达出来，以便用户查看，从而确定需求分析员的理解是否正确。“表达需求”的关键是用什么样的工具描述对系统的理解，要让用户和其他软件开发人员都能够看得懂。只有用户能够看懂逻辑模型，才能够与需求分析员一起探讨和修改需求分析；而系统设计员和程序员也只有能够正确地理解逻辑模型，才能够保证开发出的目标系统符合用户的需求。

3.1.2 需求分析的重要性

软件开发的宗旨就是满足用户的需求，而需求分析就是分析软件用户的需求是什么，因此，这

一阶段的任务极其重要。需求分析应该从理解用户需求出发,就软件功能与客户达成一致,估计软件风险和评估项目代价,最终形成开发计划并文档化。

如果投入大量的人力、物力、财力和时间开发一个软件产品,但由于需求分析不足导致开发出的产品没有市场,那么所有的投入都是徒劳的。如果花费很大的精力开发一个软件产品,最后却不满足用户的要求,导致要从头开始修改或干脆重新开发系统,那么这种损失对软件公司来说无疑是一个重创。

例如,1994年秋天,迪斯尼公司发布了第一个面向儿童的多媒体光盘游戏——狮子王动画故事书(The Lion King Animated Storybook)。尽管已经有许多其他公司在儿童游戏市场上运作多年,但是这次是迪斯尼公司首次进军这个市场,所以进行了大量促销宣传。结果,销售额非常可观,该游戏成为孩子们那年节假日的“必买游戏”。然而后来却飞来横祸。12月26日,圣诞节的后一天,迪斯尼公司的客户支持电话开始响个不停。很快,电话支持技术人员就淹没在来自于愤怒的家长并伴随着玩不成游戏的孩子们哭叫的电话之中。报纸和电视新闻进行了大量的报道。

后来证实,迪斯尼公司未能对市面上投入使用的许多不同类型的PC机型进行广泛的测试。软件在极少数系统中工作正常(例如在迪斯尼程序员用来开发游戏的系统中),但在大多数公众使用的系统中却不能运行。

仔细思考上述实例,可以分析出这样的事实,如果在需求分析阶段能够对软件产品的运行环境做出很好的调查和定位,就不会产生这样严重的后果。可见,需求分析的任务在整个软件开发过程中是极其重要的。

3.1.3 需求分析的任务

需求分析的主要任务就是解决所要开发的应用系统将要“做什么”的问题。需求分析就是要全面地掌握用户的各项要求,准确地表达出用户的实际需求,从而确定系统必须实现的功能。

需求分析不是确定系统如何完成它的工作,而是确定系统必须完成哪些工作,也就是对目标系统提出完整、准确、清晰、具体的要求。在这个阶段结束时交出的文档中应该包括详细的数据流程图(DFD)、数据字典(DD)和一组简明的算法描述。

另外,需求分析的结果是系统开发的基础,关系到系统开发的成功与否以及软件产品的质量。因此,还必须用行之有效的方法对软件需求进行严格的审查验证。

一般来说,需求分析阶段的具体任务包括以下几个方面:

(1) 确定对系统的综合需求。

对系统的综合需求主要有4个方面:

- 系统功能需求。应该划分出系统必须完成的所有功能。
- 系统性能需求。系统所需的内存容量、后援存储、重新启动和安全性能等都属于性能要求。
- 运行需求。主要指对系统运行时所处环境的要求。
- 将来可能提出的需求。应该明确那些虽然不属于当前系统开发范畴,但是根据分析将来可能会提出的要求。

(2) 分析系统的数据需求。

分析系统的数据需求是由系统的信息流归纳抽象出数据元素组成、数据的逻辑关系、数据字典格式和数据模型,并以输入/处理/输出(IPO)的结构方式表示。因此,必须分析系统的数据需求,

这是软件需求分析的一个重要任务。

(3) 导出系统的逻辑模型。

理解当前系统“怎样做”的基础上，抽取其“做什么”的本质。明确目标系统要“做什么”，可以导出系统的详细逻辑模型。

具体做法是：

- 确定目标系统与当前系统的逻辑差别。
- 将变化部分看做是新的处理步骤，对功能图（一般为数据流图）及对象图进行调整。
- 由外及里对变化的部分进行分析，推断其结构，获得目标系统的逻辑模型。通常用数据流图、数据字典和主要的处理算法描述这个逻辑模型。

(4) 修正系统开发计划。

在分析过程中，需求分析员对目标系统有了更深入、更具体的认识，因此可以对系统的成本和进度做出更准确的估计，在此基础上应该对开发计划进行修正。

(5) 利用原型化方法开发原型系统。

原型化方法就是尽可能快地建造一个粗糙的系统，这个系统实现了目标系统的某些或全部功能，但是这个系统可能在可靠性、界面的友好性或其他方面上存在缺陷。

建造原型系统的目的如下：

- 检验系统的关键设计方案的可行性。例如，算法的可行性分析、技术的可行性分析。
- 考察系统是否真正满足用户的需求。例如，为了考察是否满足用户的要求，可以用某些软件工具快速地建造一个原型系统，这个系统只是一个界面，然后听取用户的意见，改进这个原型，以后的目标系统就在原型系统的基础上开发。

在软件开发过程中，通过原型系统的使用，用户基于实践获得了关于未来的系统将怎样为他们工作的更直接的概念，从而可以更准确地提出和确定他们的要求。

可见，原型系统就是软件的一个早期可运行的版本，它实现了目标系统的某些或全部功能。用户试用了原型系统可以指出系统的哪些特性是受欢迎的，哪些特性是他们感到不可取的，以及还需要增加哪些特性。根据用户试用的实际情况，需求分析员不断改进需求内容。这种基于实践检验过的用户需求而开发出来的系统才能真正满足用户的需求。

当前，在软件开发中采用原型系统策略的主要困难是成本问题。

3.2 需求分析的过程

3.2.1 需求分析的过程

通常，把整个软件需求工程划分为需求开发和需求管理两个部分，如图 3-2 所示。

1. 需求开发阶段

需求开发阶段的工作可以分为 4 个方面：问题获取、分析、编写规格说明、验证。

(1) 问题获取。

问题识别就是从系统角度来理解软件，确定对所开发系统的综合需求，并提出这些需求的实现

条件, 以及需求应该达到的标准, 从而确定产品所期望的用户类别, 获取每个用户类的需求, 了解实际用户任务和目标以及这些任务所支持的业务需求。

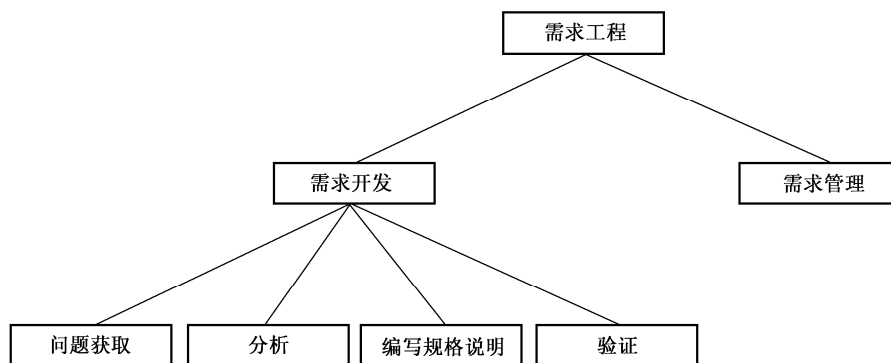


图 3-2 需求工程

这些需求包括:

- 功能需求 (做什么)。
- 性能需求 (要达到什么指标)。
- 环境需求 (如机型、操作系统等)。
- 可靠性需求 (不发生故障的概率)。
- 安全保密需求。
- 用户界面需求。
- 资源使用需求 (软件运行时所需的内存、CPU 等)。
- 软件成本消耗与开发进度需求。
- 预先估计以后系统可能达到的目标。

(2) 分析。

分析就是逐步细化所有的软件功能, 找出系统各元素间的联系、接口特性和设计上的限制, 分析他们是否满足需求, 剔除不合理部分, 增加需要部分, 最后综合成系统的解决方案, 给出要开发的目标系统的逻辑模型 (做什么的模型)。

(3) 编写规格说明书。

编写规格说明书即编制需求文档, 描述需求的文档称为软件需求规格说明书。需求开发阶段的成果之一就是需求规格说明书 (需要向下一阶段提交)。

(4) 验证。

验证主要指验证需求规格说明书, 确保对用户需求达到共同的理解与认识, 并在整个开发小组接受说明之前将问题都解决。

2. 需求管理阶段

需求管理需要 “建立并维护在软件工程中同客户达成的合同” (在目前的软件开发领域主要是 “按契约 (合同) 编程”), 这种合同都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他们并真正把需求应用到产品中。

通常的需求管理活动如下:

- 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项目。
- 以一种可控制的方式将需求变更融入到项目中。
- 使当前的项目计划与需求一致。
- 估计变更需求所产生的影响并在此基础上协商新的承诺,这种承诺具体体现在项目解决方案上。
- 让每项需求都能与其对应的设计、源代码和测试用例联系起来以实现跟踪。
- 在整个项目过程中跟踪需求状态及其变更情况。

3.2.2 需求分析的注意事项

优秀的软件产品是建立在优秀的需求基础之上的,而高质量的需求来源于客户与开发人员之间有效的交流与合作。在需求分析过程中,客户或客户代理人 与开发人员建立良好的关系极其重要,当遇到问题时应该通过各种手段达成共识。如果遇到大的分歧,应该通过协商达成对各自责任的相互理解,消除摩擦,避免一方要求而另一方不愿意或不能够满足的需求。

只有当双方参与者都明白要成功自己需要什么,同时也知道要成功合作方需要什么时,才能建立起一种合作关系。由于项目压力与日俱增,所有风险承担者有着一个共同的目标这一点容易被遗忘。其实客户和开发人员都想开发出一个既能实现商业价值,又能满足用户需要,还能使开发人员感到满足的优秀软件产品。

软件客户需求权利书列出了 9 条关于客户在项目需求工程实施中与分析人员、开发人员交流时的合法要求。每一项权利都对应着软件开发人员、需求分析人员的义务。而软件客户需求义务书也列出了 10 条关于客户在需求过程中应承担的义务。

1. 客户的权利

(1) 要求分析人员使用符合客户语言习惯的表达。

需求讨论应集中于业务需要和任务,故要使用业务术语。客户应将有关术语(例如采价、印花商品等采购术语)教给分析人员,而不一定要懂得计算机行业的术语。

(2) 要求分析人员了解客户的业务及目标。

通过与用户交流来获取用户需求,分析人员才能更好地了解客户的业务和如何才能使产品更好地满足客户的需要,这将有助于开发人员设计出真正满足客户的需要并达到其期望的优秀软件。

(3) 要求分析人员编写软件需求规格说明书。

分析人员要把从客户那里获得的所有信息整理,以区分开业务需求及规范、功能需求、质量目标、解决方法和其他信息。通过这些分析就能得到一份软件需求分析规格说明书,而这份软件需求分析规格说明书便在开发人员和客户之间针对要开发的产品内容达成了协议。软件需求分析规格说明书可以用一种客户认为易于翻阅和理解的方式组织编写,要评审编写出的软件需求分析规格说明书以确保其准确而完整地表达了客户的需求。一份高质量的软件需求分析规格说明书能有助于开发人员开发出客户真正需要的产品。

(4) 要求得到需求工作结果的解释说明。

分析人员可能采用了多种图表作为文字性软件需求分析规格说明书的补充,因为如工作流程图之类的图表能很清楚地描述出系统行为的某些方面,所以需求说明中的各种图表有着极高的价值。虽然它们不太难以理解,但是客户很可能对此并不熟悉。因此可以要求分析人员解释说明每张图表

的作用和其他需求开发工作结果及符号的意义，以及如何检查图表有无错误或不一致等。

(5) 要求开发人员尊重用户的意见。

如果用户与开发人员之间不能相互理解，那么关于需求的讨论将会有障碍，共同合作能使大家“兼听则明”。参与需求开发过程的客户有权要求开发人员尊重他们并珍惜他们为项目成功所付出的时间，同样客户也应对开发人员为项目成功这一共同目标所做出的努力表示尊重与感激。

(6) 要求开发人员对需求及产品实施提供建议和主意。

通常，客户所说的“需求”已是一种实际可能的实施解决方案，分析人员将尽力从这些解决方法中了解真正的业务及其需求。同时还应找出已有系统不适合当前业务之处，以确保产品不会无效或低效。在彻底弄清业务领域内的事情后，分析人员有时就能提出相当好的改进方法，有经验且富有创造力的分析人员还能提出增加一些用户并未发现的很有价值的系统特性。

(7) 描述产品易使用的特性。

用户可以要求分析人员在实现功能需求的同时还要注重软件的易用性，因为这些易用性或质量属性能使客户更准确且高效地完成任务。

例如，客户有时要求产品要“用户友好”或“健壮”或“高效率”，但这对于开发人员来说并无实用价值。正确的做法是分析人员通过询问和调查了解客户所要的友好、健壮及高效所包含的具体特性。

(8) 调整需求，允许重用已有的软件组件。

需求通常要有一定的灵活性，分析人员可能发现已有的某个软件组件与客户描述的需求很相符。在这种情况下，分析人员应提供一些修改需求的选择以便开发人员能够在新系统开发中重用一些已有的软件。如果有可重用的机会出现，同时客户又能调整需求说明，则能降低成本并节省时间，而不必严格按原有的需求说明开发。所以如果想在产品中使用一些已有的商业常用组件，而它们并不完全适合客户所需的特性，这时一定程度上的需求灵活性就显得极为重要了。

(9) 获得满足客户功能和质量要求的系统。

每个人都希望项目获得成功，但这不仅要求客户要清晰地告知开发人员关于系统“做什么”所需的所有信息，而且还要求开发人员能通过交流了解取舍与限制，一定要明确说明客户的假设和潜在的期望，否则开发人员开发出的产品很可能无法让客户满意。

2. 客户的义务

(1) 给分析人员讲解业务。

分析人员要依靠客户给他们讲解的业务概念及术语，但客户不能指望分析人员会成为该领域的专家，而只能让他们真正明白客户的问题和目标。不要期望分析人员能把握客户业务的细微之处，他们很可能并不知道那些对于客户来说理所当然的常识。

(2) 抽出时间清楚地说明并完善需求。

客户很忙，经常在最忙时还得参与需求开发。但无论如何，客户有义务抽出时间参与需求讨论，接受采访或其他获取需求的活动。有时分析人员可能以为明白了客户的观点，而过后发现还需要客户的讲解，这对软件产品的成功极为重要。

(3) 准确而详细地说明需求。

编写一份清晰而准确的需求文档是很困难的，由于处理细节问题不但烦人，而且耗时，因此很容易留下模糊不清的需求。但是在开发过程中必须解决这种模糊性和不准确性，在软件需求分析规格说明书中暂时加上 TBD (to be determined) 标志或汉字“待定”。用该标志可指明哪些是需要进

一步探讨、分析或增加信息的地方。不过，有时也可能因为某个特殊需求难以解决或没有人愿意处理它而注上 TBD 标志。

尽量将每项需求的内容都阐述清楚，以便分析人员能准确地将其写进软件需求分析规格说明书中。如果一时不能准确表述，则允许获取必要的准确信息这样一个过程。通常使用所谓的原型技术，通过开发的原型系统，客户可以同开发人员一起反复修改并不断完善需求定义。

(4) 及时地做出决定。

分析人员将要求客户做出一些选择和决定，这些决定包括来自多个用户提出的处理方法或在质量特性冲突和信息准确度中选择折衷方案等。有权做出决定的客户必须积极地对待这一切，尽快做出处理。因为开发人员通常只有等客户做出了决定才能行动，而这种等待会延误项目的进展。

(5) 尊重开发人员的需求可行性及成本评估。

所有的软件功能都有其成本价格，开发人员最适合预算这些成本（尽管许多开发人员并不擅长评估预测）。客户所希望的某些产品特性可能在技术上行不通，或者实现它要付出极为高昂的代价。而某些需求试图在操作环境中要求不可能达到的性能或试图得到一些根本得不到的数据，开发人员会对此做出负面的评价意见，客户应该尊重他们的意见。有时，客户可以重新给出一个在技术上可行且实现上便宜的需求。

例如，要求某个行为在“瞬间”发生是不可行的，但换种更具体的时间需求说法（在 100ms 以内）即可实现。

(6) 划分需求优先级别。

大多数项目没有足够的时间或资源来实现功能性的每个细节，决定哪些特性是必要的、哪些是重要的、哪些是好的，是需求开发的主要部分。只能由客户来负责设定需求优先级，因为开发人员并不可能按客户的观点决定需求优先级，开发人员将为客户确定优先级提供有关每个需求的花费和风险的信息。

(7) 评审需求文档和原型。

无论是正式的还是非正式的方式，对需求文档进行评审都会对软件质量的提高有所帮助。让客户参与评审才能真正鉴别需求文档是否完整并正确地说明期望的必要特性。评审也给客户代表提供一个机会，给需求分析人员带来反馈信息以改进他们的工作。

客户能提供有价值的反馈信息给开发人员，帮助他们更好地理解需求。必须认识到，原型并不是一个实际产品，开发人员能将其转变扩充为功能齐全的系统。

(8) 需求出现变更要立即联系。

不断的需求变更会给在预定计划内完成高质量产品带来严重的负面影响。变更是不可避免的，但在开发周期中变更越在晚期出现，其影响越大。变更不仅会导致代价极高的返工，而且工期也会被迫延误，特别是在大体结构已完成后又需要增加新特性时。所以一旦客户发现需要变更需求时，请一定立即通知分析人员。

(9) 应遵照开发组织处理需求变更的过程。

为了将变更带来的负面影响减少到最低限度，所有的参与者必须遵照项目的变更控制过程。这要求不放弃所有提出的变更，对每项要求的变更进行分析和综合考虑，最后做出合适的决策以确定将哪些变更引入项目中。

(10) 尊重开发人员采用的需求工程过程。

软件开发中最具挑战性的莫过于收集需求并确定其正确性，分析人员采用的方法有其合理性。

也许客户认为需求过程不太划算,但请相信花在需求开发上的时间是“很有价值”的。如果客户理解并支持分析人员为收集编写需求文档和确保其质量所采用的技术,那么整个过程将会更为顺利。

3.2.3 需求风险

重视需求过程是非常重要的,即使这样,需求过程自身的缺陷也有可能给系统开发的成功带来某些风险,这里的“成功”是指推出的产品能以合理的价格、完善的功能以及可靠的质量完全满足用户的期望。下面将讨论一些需求风险,在软件开发过程中同样值得注意。

1. 没有足够的用户参与

客户经常不明白为什么收集需求和为确保需求质量需要花费那么多工夫,开发人员可能也不重视用户的参与。究其原因,一是因为开发人员感觉与用户合作不如编写代码有意思;二是因为开发人员认为已经明白用户的需求了。在某些情况下,与实际使用产品的用户直接接触很困难,而客户也不太明白自己的真正需求。但还是应让具有代表性的用户在项目早期直接参与到开发队伍中,并一同经历整个开发过程。

系统人员在实践过程中也有些感觉,在实施一个项目时,若无足够的用户参与,系统人员获得的需求是片面且不完整的,这样系统在需求之初就埋下风险。

2. 用户需求的不断增加

在开发中若不断地补充需求,项目就越变越庞大,从而导致超过其计划及预算范围。计划并不总是与项目需求规模、复杂性、风险、开发生产率及需求变更实际情况相一致,这使得问题更难解决。实际上,问题根源在于用户需求的改变和开发人员对新需求所做的修改。

3. 模棱两可的需求

模棱两可是需求规格说明中最为可怕的问题,它的一层含义是指诸多用户对需求说明产生了不同的理解;另一层含义是指单个用户能用不止一种方式来解释某个需求说明。

模棱两可的需求会使不同的风险承担者产生不同的期望,它会使开发人员为错误问题而浪费时间,并且使测试者与开发人员所期望的不一致。在实际开发过程中,如果测试者对需求产生错误理解,则会导致测试者不得不重写许多测试用例并重做许多测试,从而延误了开发进程。

处理模棱两可需求的一种方法是组织好负责从不同角度审查需求的队伍,仅仅简单浏览一下需求文档是不能解决模棱两可问题的。如果不同的评审者从不同的角度对需求说明给予解释,并且每个评审人员都真正了解需求文档,这样二义性就不会直到项目后期才被发现,那时再发现的话会使得更正代价很大。

4. 不必要的特性

“画蛇添足”是指开发人员力图增加一些“用户欣赏”,但需求规格说明书中并未涉及的新功能。经常发生的情况是用户并不认为这些功能很有用,以致在其上耗费的努力失去了作用。开发人员应当为客户构思方案并为他们提供一些具有创新意识的思路,具体提供哪些功能要在客户所需与开发人员在允许时限内的技术可行性之间求得平衡。开发人员应努力使功能简单易用,而不要未经客户同意擅自脱离客户要求,自作主张。

同样,客户有时也可能要求一些缺乏实用价值的功能,而实现这些功能只能徒耗时间和成本。

为了将“画蛇添足”的危害尽量减小,开发人员应明确所包括的功能是否恰当,以及这些功能的“来龙去脉”,这样使得需求分析过程始终是注重那些能使用户完成其业务任务的核心功能。

5. 忽略了用户分类

大多数产品是由不同的人使用其不同的特性,使用者受教育程度和经验水平不尽相同,使用不同特性的频繁程度也有所差异。如果不能在项目早期针对所有这些主要用户进行分类的话,必然导致有的用户对产品感到失望。例如菜单驱动操作对高级用户太低效了,但含义不清的命令和快捷键又会使不熟练的用户感到困难。

6. 不准确的需求目标

据统计,导致需求过程中软件成本估计极不准确的原因主要有频繁的需求变更、遗漏的需求、与用户交流不够、质量低下的需求规格说明和不完善的需求分析等。

对于不准确的需求的正确回应是:“等我真正明白客户的需求时,我就会来告诉客户”。基于不充分信息和未经深思的对需求不成熟的估计很容易被一些因素左右,要做出估计时,最好还是给出一个范围。未经准备的估计通常是作为一种猜测给出的,可行性并不确定。因此作为软件开发人员要尽力做出可达到的目标并坚持完成它。

3.3 数据流程图

数据流程图(Data Flow Diagram, DFD)是一种图形化技术,它描绘信息流和数据流从输入口转到输出的过程中所经历的变换,既提供了功能建模机制,又提供了信息建模机制。

3.3.1 数据流程图相关图示

1. 数据流程图的基本图形符号

数据流程图的基本图形符号如图3-3所示。

图标说明:

- 正方形(或立方体):表示数据的源点或终点。
- 圆角矩形(或椭圆形):代表变换数据的处理。
- 开口矩形(或两条平行横线):代表数据存储。
- 水平箭头(垂直箭头):表示数据流,即特定数据的流动方向。

2. 软件系统中的数据流程图

一个完整的软件系统包括系统的外部实体、数据处理、数据存储和系统中的数据流4个组成部分,如图3-4所示。

(1) 外部实体。

外部实体指系统以外和系统有联系的人或事物,它说明了数据的外部来源和去处,属于系统的外部或系统的界面。外部实体支持系统数据输入的实体称为起点,支持系统数据输出的实体称为终点。

如图3-5所示,通常外部实体在数据流程图中用正方形框(或立方体)表示,框中写上外部实体名称,为了区分不同的外部实体,可以在正方形的左上角用一个字符表示,同一外部实体可在一张数据流程图中出现多次,这时在该外部实体符号的右下角画上小斜线表示重复。

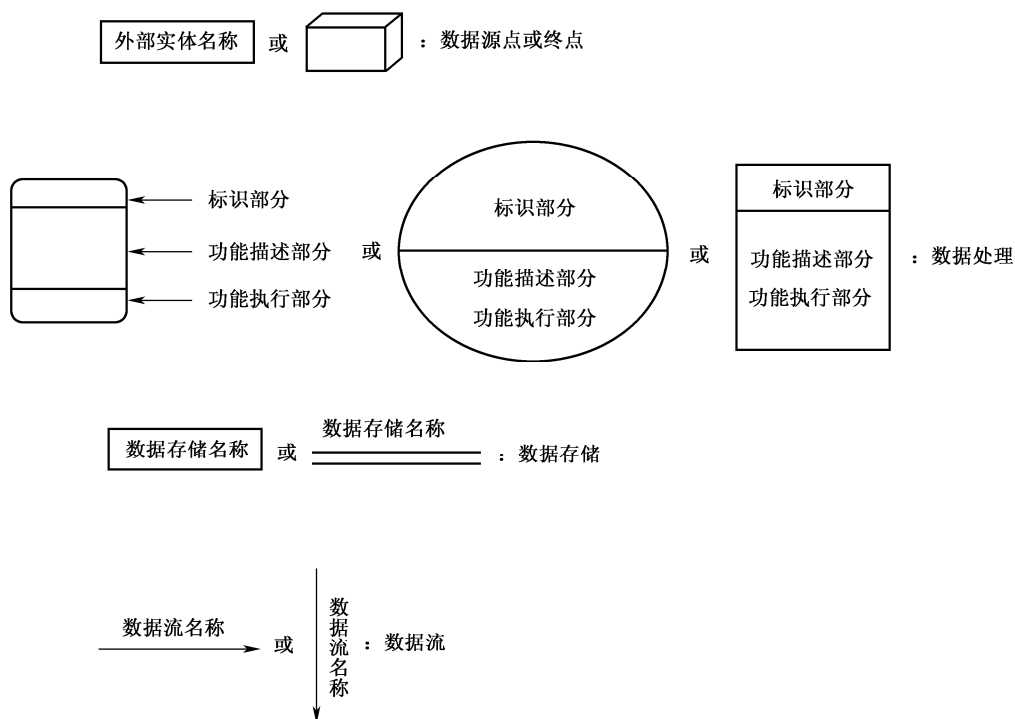


图 3-3 数据流程图的基本图形符号

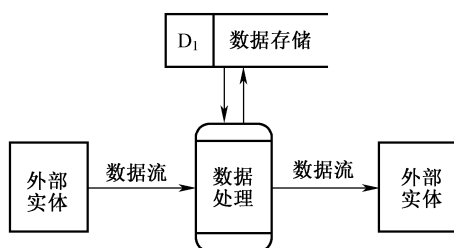


图 3-4 软件系统的组成部分

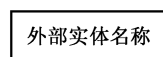


图 3-5 外部实体图示

(2) 数据处理。

处理是指对数据逻辑的处理，也就是数据变换，也称加工。它用来改变数据值。而每一个处理又包括数据输入、数据处理和数据输出等部分。

如图 3-6 所示，在数据流程图中处理过程用圆角矩形（或椭圆形）表示，圆角矩形分 3 个部分，标识部分用来标识一个功能，功能描述部分是必不可少的，功能执行部分表示功能由谁来完成。

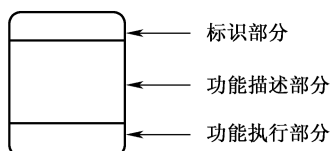


图 3-6 数据处理图示

(3) 数据存储。

数据存储表示数据保存的地方，它用来存储数据。系统处理从数据存储中提取数据，也将处理的数据返回数据存储。与数据流不同的是数据存储本身不产生任何操作，它仅仅响应存储和访问数据的要求。

如图 3-7 所示，在数据流程图中数据存储用右边开口的矩形（或两条平行横线）表示。在长方条内写上数据存储名字。为了区别和引用方便，左端加一小格，再标上一个标识，用字母 D 和数字组成。

(4) 数据流。

数据流是指处理功能的输入或输出。它用来表示某个中间数据流值，但不能用来改变数据值。数据流是模拟系统数据在系统中传递过程的工具。

如图 3-8 所示，在数据流程图中用一个水平箭头或垂直箭头表示，箭头指出数据的流动方向，箭线旁注明数据流名称。

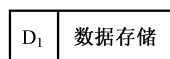


图 3-7 数据存储图示

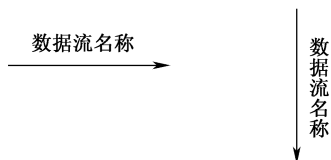


图 3-8 数据流图示

3.3.2 数据流程图的设计

1. 数据流程图的画法

画数据流程图的基本目的是利用它作为交流信息的工具。数据流程图的另一个主要用途是作为系统分析和设计的工具。

数据流程图的画法要点是先把整个系统按总的功能画出最粗的数据流程图，暂不考虑系统内部的各种信息处理、传递、存储等，只表示出系统的总体功能、系统的边界、与外界的接口等。

其画法具体步骤如下：

- (1) 确定数据流程图的总体功能。把软件系统看成一个整体功能，明确信息的输入和输出。
- (2) 找出数据流程图的起点与终点，它们是系统的边界。找到系统的外部实体，一旦找到外部实体，则系统与外部世界的界面就可以确定下来，系统的数据流的起点和终点也就找到了。
- (3) 找出外部实体中起点的输入数据流和终点的输出数据流，在图的边上画出系统的外部实体。
- (4) 从外部实体的起点输入数据流出发，按照系统的逻辑需要，逐步画出一系列逻辑处理，直到找到外部实体处理所需的终点输出数据流，形成数据流的封闭。
- (5) 再将系统内部数据处理部分看做整体功能，其内部也有信息的处理、传递、存储过程。
- (6) 按照以上 5 步，一级一级地剖析，直到所有处理步骤都被很具体地表示为止。

2. 数据流程图的设计要点

- (1) 自外向内、自顶向下、逐层细化、完善求精。
- (2) 保持父图与子图的平衡。任何一个数据流子图必须与它的父图上的一个处理过程对应，

两者的输入数据流和输出数据流必须一致，在数量和名字上相同。

(3) 保持数据守恒。即一个处理的所有输出数据流中的数据必须能从该处理的输入数据流中直接获得，或者能通过该处理产生的数据。

(4) 根据抽象原则，处理细节隐蔽，只需画出处理和处理之间的关系即可。

(5) 均匀分解，应该使一个数据流中的各个处理分解层次大致相同。

(6) 每个数据处理过程至少有一个输入数据流和一个输出数据流。

(7) 关于数据流程图元素问题。

- 数据流程图上所有图形符号必须是前面所述的 4 种基本元素。
- 数据流程图的主图必须含有前面所述的 4 种基本元素，缺一不可。
- 数据流程图上的数据流必须封闭在外部实体之间，外部实体可以是一个，也可以是多个。
- 数据流程图上的每个元素都必须有名字。

3. 数据流程图设计的注意事项

(1) 数据存储与数据流都是数据，仅仅是所处的状态不同。前者是处于静止状态的数据，而后者则是处于运动中的数据。

(2) 在数据流图中应该描绘所有可能的数据流向，而不应该描绘出现某个数据流的条件。

(3) 数据流程图的基本要点是描绘“做什么”而不考虑“怎么做”，因此通常在数据流图中忽略出错处理，也不包括诸如打开或关闭文件之类的内务处理。

(4) 检查数据流程图。对一个系统的理解，不可能一开始就完美无缺，开始分析一个系统时，尽管需求分析员对问题的理解有不正确、不确切的地方，但还是应该根据自身的理解，用数据流程图表达出来，进行核对，逐步修改，获得较为完美的流程图。

(5) 提高数据流程图的易理解性。

(6) 数据流程图是系统分析员调查业务过程，与用户交换思想的工具，因此数据流程图应简明易懂，这也有利于后面的设计，有利于对系统说明书进行维护。

4. 设计数据流程图的主要作用

(1) 便于用户表达功能需求和数据需求及其联系。

(2) 便于两类人员共同理解现行系统和规划系统的框架。

(3) 清晰表达数据流的情况。

(4) 有利于系统建模。

3.3.3 分层数据流程图

对于复杂问题的数据处理过程，可以根据其处理问题的层次关系进行逐步分解，并用分层数据流程图反映出来。

逐层扩展数据流程图，是对上一层图中某些处理框加以分解。随着处理的分解，功能越来越具体，数据存储、数据流越来越多。究竟怎样划分层次，划分到什么程度，没有绝对标准，一般认为展开的层次与管理层次一致，也可以划分得更细，处理块的分解要自然，注意功能完整性。“分层”技术就是对数据流程图逐步深入分析，加入各项细节。概括地说，就是自外向内、自顶向下、逐层细化、完善求精。

根据层次关系一般将数据流程图分为顶层数据流程图、中间数据流程图和底层数据流程图。

顶层数据流程图是从全企业的高度,综合、整体地观察每一个职能域数据流的进出概况。通过顶层数据流将一些职能域联结起来,使分析人员形成对全企业数据流的整体认识。中间层和底层数据流程图是某一职能域内部的业务过程和数据流的进一步调查的记录,关键是业务过程的识别与定义,以及存储类用户视图的定义与规范化。

如图 3-9 所示,给出了分层数据流程图的图示。数据处理 S 包括 3 个子系统 1、2、3。顶层下面的第一层数据流程图为 DFD/L1 (DFD, Data Flow Diagram)。第二层数据流程图 DFD/L2.1、DFD/L2.2 及 DFD/L2.3 分别是子系统 1、2 和 3 的细化。对任何一层数据流程图来说,我们称它的上层图为父图,在它下一层的图则称为子图。在数据处理过程中从 F 中读取数据或向 F 输出数据。

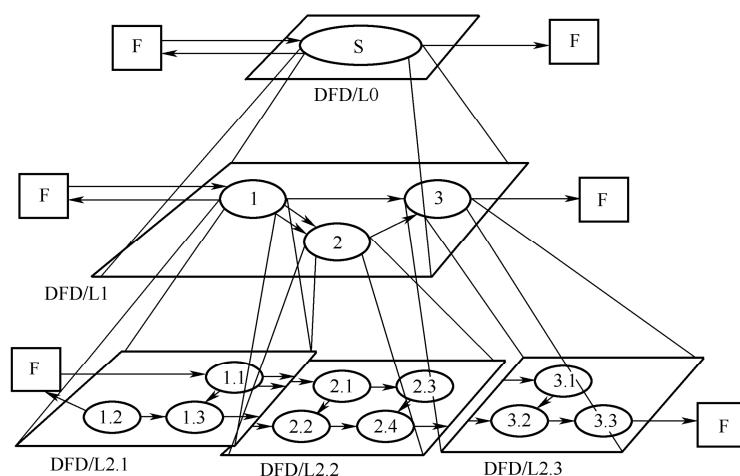


图 3-9 分层数据流程图的图示

如图 3-10 所示,给出了分层数据流程图的实例。

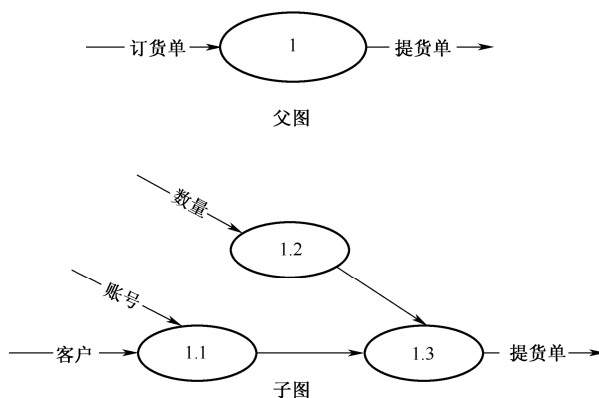


图 3-10 分层数据流程图的实例

3.4 数据字典

数据流程图描述了软件系统的组成及各部分之间的联系,但没有定义各部分的具体定义。只有

为流程图中各个组成元素做出明确定义后，才能完整地描述一个软件系统。

数据字典（Data Dictionary, DD）就是对数据流程图中出现的所有被命名的组成元素作为条目加以定义，使得每一个组成元素的名字都有一个确切的解释。所有的条目按一定次序排列，构成一本数据字典，提供给开发人员和用户查阅。

数据字典是对数据流程图的补充说明，它的编制和维护是一项非常繁重的工作，一旦建立起来，从系统分析直至系统运行都要用到它。数据流程图和数据字典是需求规格说明书的主要部分。

数据字典中有 4 类条目：数据流（Data Stream）、数据元素（Data Element）、数据存储（Data Store）和数据处理过程（Data Processing）。

1. 数据流条目

数据流条目给出某个数据流的定义，列出该数据流的各个组成数据项。

描述数据流的有关属性如下：

- 编号。
- 数据流名称。
- 简述：简要介绍作用，即它产生的原因和结果。
- 数据流来源：来自何方。
- 数据流去向：去向何处。
- 数据流组成：数据流所包含的数据结构。
- 数据流的流通量：单位时间里的数据传输次数。
- 高峰流量。

2. 数据元素条目

数据元素条目又称数据项或数据流分量，主要是给出某个数据单项的定义，通常是该数据项的值类型、允许值等。

具体内容如下：

- 编号。
- 数据元素名称。
- 类型。
- 长度。
- 取值范围。
- 相关的数据元素及数据结构。

3. 数据存储文件条目

数据存储文件条目给出某个文件的定义，列出了其记录的组成数据项以及文件的组织方式。

描述数据存储文件条目的主要属性如下：

- 编号。
- 数据存储名称。
- 简述：存放的是什么数据。
- 输入数据。
- 输出数据。
- 数据存储组成：数据结构。
- 存储方式：顺序、直接、关键码。

- 存储频率。

4. 数据处理条目

数据处理，也称数据加工，描述数据流的处理（加工）过程。

描述数据处理条目的主要属性如下：

- 编号。
- 数据处理名称。
- 输入数据。
- 输出数据。
- 关联文件。
- 处理逻辑。

3.5 需求规格说明书

1. 需求规格说明书的必要性

需求规格说明书是需求分析阶段必须具备的成果。如果因为忽略需求文档而导致重复返工，其后果将非常严重。因为重新编制代码的代价远远超过重写一份需求文档的代价。

需求规格说明书是基于软件合同或立项建议书以及分析人员对用户现场的调研，经过分析协商，生成的最终相关需求文档。

既然需求规格说明书是软件生命周期中一份至关重要的文档，在需求分析阶段必须及时建立并保证其质量。需求规格说明书实际上是为目标系统设计的相关逻辑模型，利用数据流程图、数据字典等描述方法为开发早期尚未形成的软件产品建立一个可视化的总体模型。

2. 需求规格说明书的内容

需求规格说明书是描述软件各项规格的，一般应该主要包括以下7个方面的内容：

（1）概述。

- 需求分析的原则。
- 需求分析的方法。

（2）系统功能。

- 功能描述。
- 数据流程图。

（3）数据字典。

- 数据元素。
- 数据流。
- 数据存储；
- 外部项。

（4）小说明。

小说明是用来描述数据处理（也称加工）的。在一个分层数据流程图中，上层的加工通过细化分解为下层的加工。

（5）数据量估计。

- 数据容量总计。
- 数据的分布和传输。
- (6) 数学模型及其说明。
- (7) 开发及运行环境设置。

说明：需求规格说明书是由一个基础框架和一系列相关条目的描述组成的。对于不同的目标系统，需求规格说明书的框架可以有所不同，但原则都是一样的，就是能够完整、准确地表达出用户和软件开发人员对系统需求的共同理解。简单易懂、易于修改是任何需求规格说明书必须具备的特点。

3. 需求规格说明书的编写

编写需求规格说明书的方法如下：

- (1) 使用好的结构化的自然语言编写文本型文档。
- (2) 建立图形化逻辑模型，这些逻辑模型可以描绘转换过程、系统状态及其之间的变化、数据关系，以及逻辑流或对象类及其关系。
- (3) 编写形式化规格说明，这可以通过使用数学上精确的形式化逻辑语言来定义需求。

由于形式化规格说明具有很强的严密性和精确度，因此所使用的形式化语言只有极少数软件开发人员才熟悉，更不用说客户了。虽然结构化的自然语言具有许多缺点，但在大多数软件工程中，它仍是编写需求文档最现实的方法，包含功能和非功能需求的基于文本的软件需求分析规格说明书已经为大多数项目所接受。图形化分析模型通过提供另一种需求视图，增强了软件需求分析规格说明书。

软件需求分析规格说明书不仅是系统测试和用户文档的基础，也是所有子系列项目规划、设计和编码的基础，它应该尽可能完整地描述系统预期的外部行为 and 用户可视化行为。除了设计和实现上的限制，软件需求分析规格说明书不应该包括设计、构造、测试或工程管理的细节。

软件需求分析规格说明书作为产品需求的最终成果必须具有综合性，必须包括所有的需求，开发人员和客户不能做任何假设。如果任何所期望的功能或非功能需求未写入软件需求分析规格说明书，那么它将不能作为协议的一部分并且不能在产品中出现。

为了编写好软件需求规格说明书，要注意以下几点：

- (1) 对节、小节和单个需求的号码编排必须一致。
- (2) 在右边部分留下文本注释区。
- (3) 允许不加限制地使用空格。
- (4) 正确使用各种可视化强调标志（例如黑体、下划线、斜体和其他不同字体）。
- (5) 创建目录表和索引表，有助于读者寻找所需的信息。
- (6) 对所有图和表指定号码和标识号，并且可按号码查阅。
- (7) 使用字处理程序中交叉引用的功能来查阅文档中其他项或位置，而不是通过页码或节号。

说明：需求规格说明书的具体样式见实训项目 3-2。

4. 需求规格说明书的作用

- (1) 作为用户和软件开发商之间的合同，为双方建立一个系统需求的文字化说明。
- (2) 反映问题的层次和结构，为系统设计和编码阶段提供参考依据。
- (3) 作为软件测试、系统验收以及实施的依据。



实训 3-1 数据字典的设计与定义

【实训目标】

掌握数据字典的定义方法。

【实训要求】

针对某个系统模块，学生能够较准确地定义相关的数据字典。

【实训内容】

某学生信息管理系统数据字典示例

(1) 数据流（Data Stream）定义表。

数据流定义表如表 3-1 所示。

表 3-1 数据流定义表

编号	数据流名称	说明	数据流组成	流通量 (次/月)	备注
S1	学生情况	-	E02+E03+E04+E05+E06+E07	-	
S2	学生分数	-	E01+E02+ E08+E09+E10	-	
S3	班级分类	-	E01+E02+E03+E04+ E05+E06+E07+E08	-	
S4	各科成绩	-	E01+E09+E010	-	
S5	课程成绩	-	E01+E02+E03+E08+E09+E10	-	
S6	查询结果	-	S04 S05	-	
S7	统计	-	S04+S05	-	

(2) 数据元素（Data Element）定义表

数据元素定义表如表 3-2 所示。

表 3-2 数据元素定义表

编号	数据元素名称	类型	长度	值域	备注
E01	学生学号	int	-	-	
E02	学生姓名	nchar	20	-	
E03	学生性别	char	2	F/M	
E04	出生日期	smalldatetime	-	-	
E05	家庭住址	nchar	50	-	

续表

编号	数据元素名称	类型	长度	值域	备注
E06	政治面貌	nchar	20	-	
E07	联系电话	char	20	-	
E08	所在班级	nchar	20	-	
E09	课程名称	nchar	20	-	
E10	课程成绩	smallint	-	0~100	

说明：使用 int、smallint、smalldatetime 等数据类型时，用户不能指定其长度，因为其长度都已经被系统设定好了。

（3）数据存储文件（Data Store File）定义表。

数据存储文件定义表如表 3-3 所示。

表 3-3 数据存储文件定义表

编号	文件名	数据存储组成	存储方式	存储频率 (次/天)	备注
F01	学生档案	{E01+E02+E03+E04+E05+E06+E07+E08+E10}	E01/升序	60	
F02	学生成绩	{E01+E02+E08+E09+E10}	E01/升序	60	

（4）数据处理（Data Processing）定义表。

数据处理定义表如表 3-4 所示。

表 3-4 数据处理定义表

编号	处理名称	输入数据	输出数据	关联文件	处理逻辑	备注
P1.1	添加	S01	S01	F01	IF 新生信息不存在 DO P1.1 ENDIF	
P1.2	删除	S01		F01	IF S01 要改动 DO P1.2 ENDIF	
P1.3	班级管理	S01	S03	空	从 P1.1 中读取添加学生信息 IF 满足班级分类条件 DO P1.3 ELSE 显示“不够条件” ENDIF	
P2.1	添加	S02	S04	F02	添加新的成绩	

续表

编号	处理名称	输入数据	输出数据	关联文件	处理逻辑	备注
P2.2	删除	S07		F02	IF S01 要改动 DO P2.2 ENDIF	
P2.3	课程管理	S04	S05	空	从 S04 中读取各科成绩信息，根据 F01 进行课程分类管理	
P3	统计	S05	S07	F02	从 F02 中读取数据，生成统计结果	

实训 3-2 “图书馆书目查询管理系统”需求分析设计

【实训目标】

掌握需求分析的设计方法。

【实训要求】

针对某个系统模块，学生能够编写出较为合理的需求规格说明书。

【实训内容】

针对实例“图书馆书目查询管理系统”，设计相关的需求规格说明书。

图书馆书目查询管理系统需求规格说明书

1.1 图书馆书目查询管理系统的背景分析

1.1.1 目标系统的基本任务

沈阳师范大学图书馆不仅拥有良好的馆舍环境和条件，还拥有丰富的实体资源和网上虚拟资源。近几年来，图书馆始终把馆藏建设作为图书馆发展和一切工作的重心，不断加大经费投入力度，不断提高馆藏质量。到目前为止，馆藏图书近 134.2 万册，年订购中外文期刊 1760 余种；电子图书 33 万余种，中外文电子期刊 2 万余种，博硕士论文 45 万篇；各种多媒体音像资料 1.2 万多种；大型网络数据库 23 个，自建特色数据库 15 个。初步形成了一个以文科为主，各学科共同发展的馆藏体系，即社会科学比重大、质量高；教育科学、汉语言文学方面的文献比较齐全；古籍及自然科学文献具有一定规模。

根据上述情况，图书馆需要建立一套网络化的书目查询管理系统，以方便广大师生查询书目信息、获取新书通报情况等，也方便图书馆的管理人员对读者情况和图书情况进行有效的电子化管理。

该图书馆书目查询管理系统的服务对象分成两类：读者和管理员。而读者又分为一般读者和注册用户。一般读者经过注册后成为注册用户，注册用户可以登录“我的图书馆”，进行读者信息查询和修改、读者密码修改、图书借阅、查看借阅历史等特殊操作。一般读者只能进行书目查询、浏览新书通报、分类浏览等普通操作。

1.1.2 图书馆机构设置图以及职能分配情况

为了便于目标系统设计，首先要对图书馆内部的组织结构以及人员分布情况有所了解。图书馆机构设置情况如图 3-11 所示。

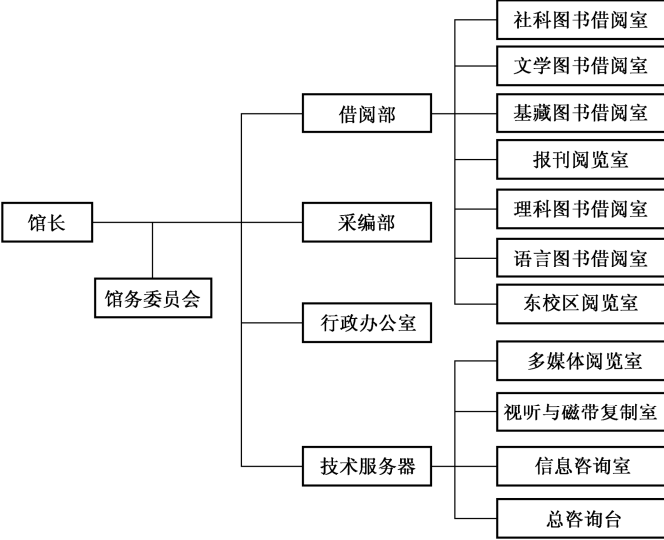


图 3-11 图书馆机构设置

各部门业务职能分配情况如表 3-5 所示。

表 3-5 图书馆机构职能分配表

机构名称	职能
馆长	主持全馆工作，领导制订发展规划、规章制度、工作计划、人员聘任及经费预算，并组织贯彻实施。副馆长和馆长助理协助馆长工作
馆务委员会	由图书馆党政领导班子成员、工会主席、各部主任组成。馆务委员会对全馆发展建设及行政业务工作中的重大问题进行商议，并做出决策
行政办公室	办公室负责全馆的行政管理、业务协调、馆际交流与合作、对外联系和接待参观来访等工作
借阅一部	主要负责入藏图书资料的流通阅览、剔除和组织管理及读者信息库的维护
借阅二部	负责报刊阅览室、理科图书借阅室、语言图书借阅室、东校区借阅室入藏图书资料的流通、阅览工作
技术服务部	负责图书馆内自动化管理系统的支持、维护与改进；各种新的信息处理技术的引进和设备的安装以及技术维护、跟踪、开发与利用；图书馆主页管理与维护；与其他部门合作，支持新技术在其他部门的应用、维护和技术培训；本馆网上资源的开发、建设与更新；通过咨询部、总咨询台、网上咨询等形式为读者解答在利用图书馆过程中所遇到的各种问题；文检课教学、新生入馆培训、电子资源使用培训、信息联络员为各院系教师开展代检代查、文献传递、定题服务；馆内电子屏幕新闻发布
采编部	根据学科建设和发展规划，负责各类文献资源的采集、收登、交换、标引、编目、典藏、回溯建库等工作。

1.2 目标系统业务流程分析

目标系统的设计目的是实现图书馆相关业务流程的电子化，因此业务流程分析是需求分析的基础环节。在 1.1.2 节中对图书馆机构设置图以及职能分配情况进行了了解，但是对组织内部各部门的具体业务情况和他们之间的业务联系并不知道。

因此，需要对目标系统的业务功能和流程进行分析，而业务流程图是分析业务功能和流程的主要工具。

业务流程图反映了目标系统中各种业务往来关系、作业流程和信息流向等。

以下是目标系统的几个相关的业务流程图。图书采编业务流程图如图 3-12 所示，图书借阅业务流程图如图 3-13 所示，读者信息管理业务流程图如图 3-14 所示，图书馆相关信息设置业务流程图如图 3-15 所示。

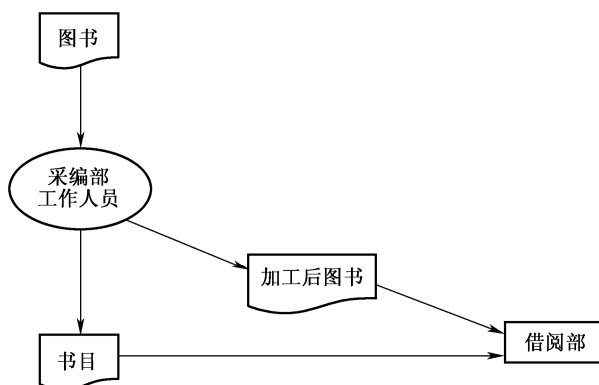


图 3-12 图书采编业务流程图

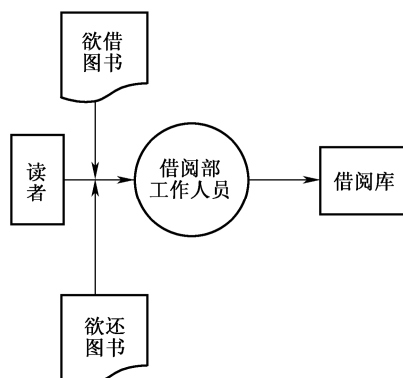


图 3-13 图书借阅业务流程图

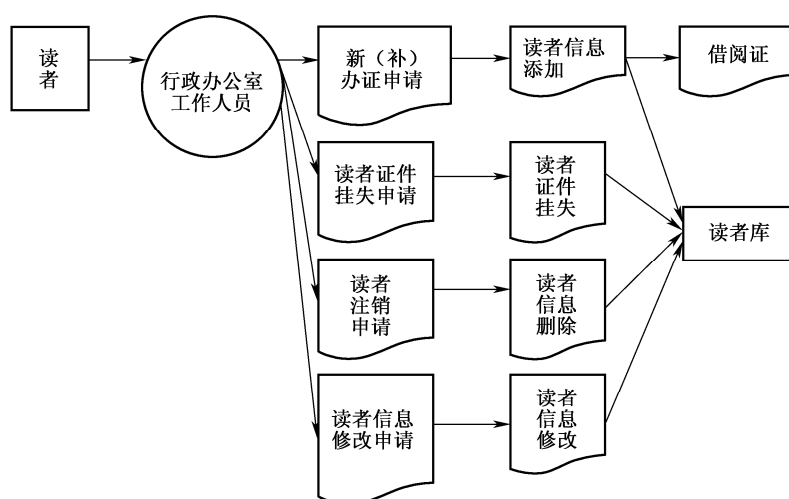


图 3-14 读者信息管理业务流程图

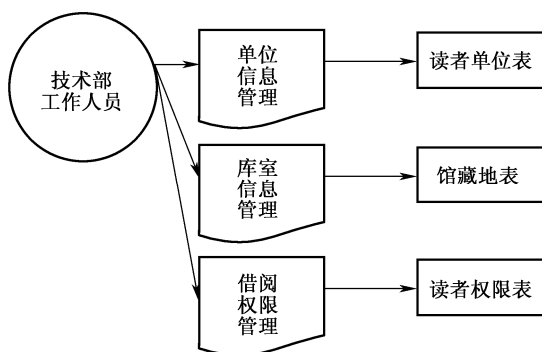


图 3-15 图书馆相关信息设置业务流程图

1.3 数据流程图

“图书馆书目查询系统”的数据流程图可以全面地描述目标系统的逻辑模型，抽象地概括了各项业务流程的关联。

1. 设计中的信息管理理念

在设计数据流程图时应该注意，目标系统不仅是传统业务流程的计算机化，更重要的是遵循科学的信息管理理念、方法和手段，不断提升图书馆电子化服务的质量和图书信息管理水平。

因此，在目标系统数据流程图的设计当中，可以看到诸如新书通报、分类浏览、借阅历史这样人性化、更贴近读者需要的服务功能；也可以看到诸如图书典藏、图书调拨这样可以提高图书管理效率的管理功能。

2. 设计中的分层技术

“图书馆书目查询管理系统”是一个较为复杂的目标系统，包含大量的数据处理任务，所以在设计中，根据其处理问题的层次关系进行逐步分解，并用分层数据流程图反映出来。

(1) 0 层数据流程图。

经过详细调查和需求分析，得到如图 3-16 所示的 0 层数据流程图。

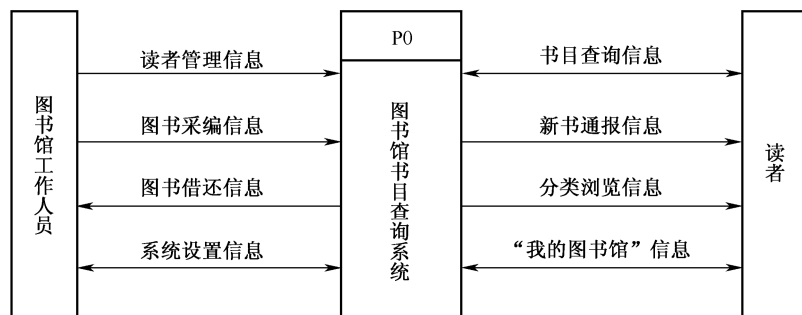


图 3-16 图书馆书目查询系统 0 层数据流程图

(2) 1 层数据流程图。

如图 3-17 所示是目标系统的 1 层数据流程图。

说明：读者库：包含读者信息表（dbo.Reader）；图书库：包含图书信息表（dbo.Marc）；借阅库：包含读者借阅历史表（dbo.LentHis）和读者借阅状态表（dbo.Lentstat）；中图法简表：即中国图书馆分类法简表。

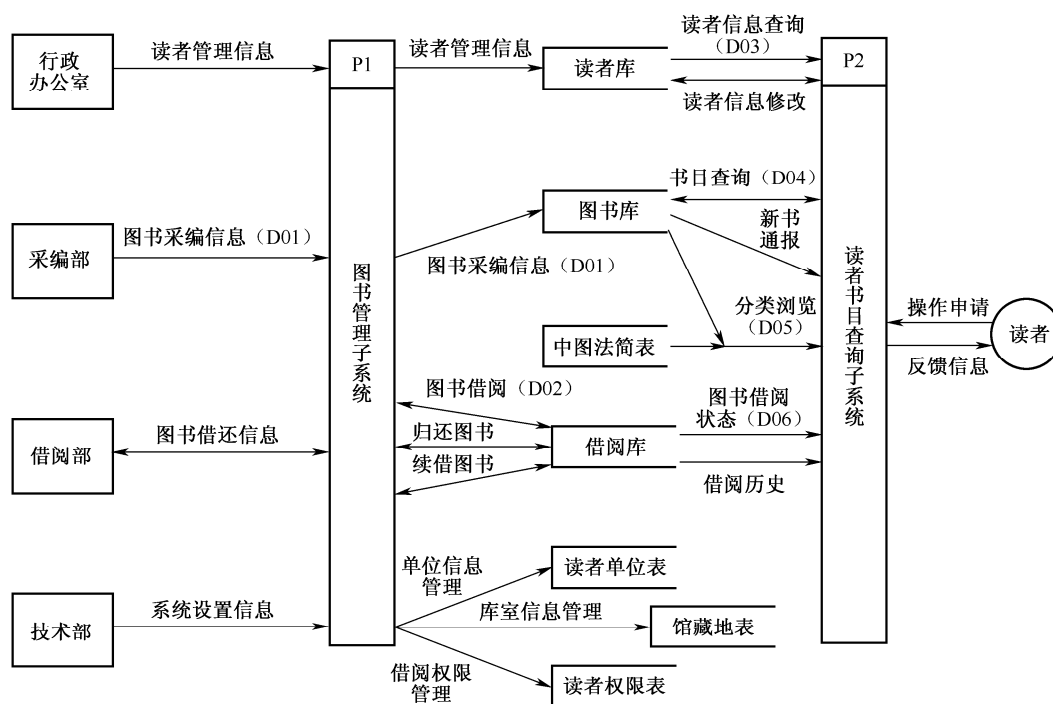


图 3-17 图书馆书目查询系统 1 层数据流程图

(3) 部分 2 层数据流程图。

1) 图书馆工作人员登录系统数据流程图如图 3-18 所示。

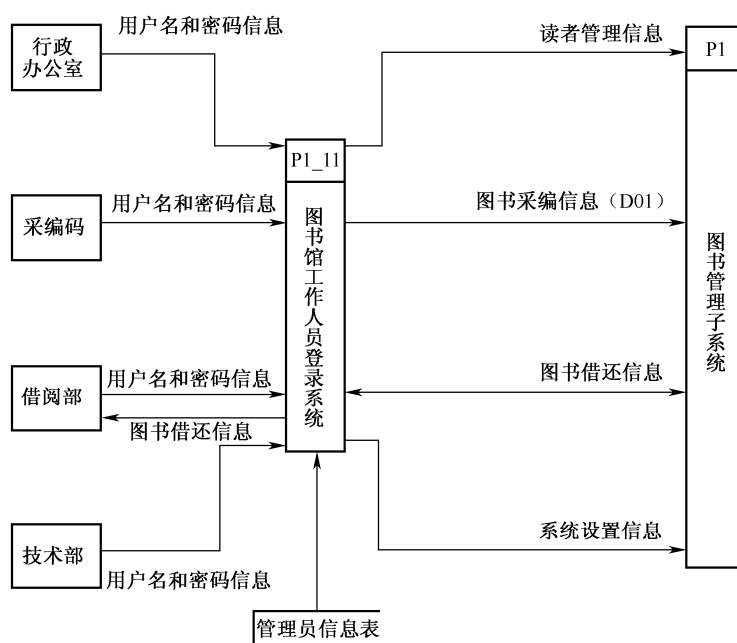


图 3-18 图书馆工作人员登录系统数据流程图

2) 读者登录系统数据流程图如图 3-19 所示。

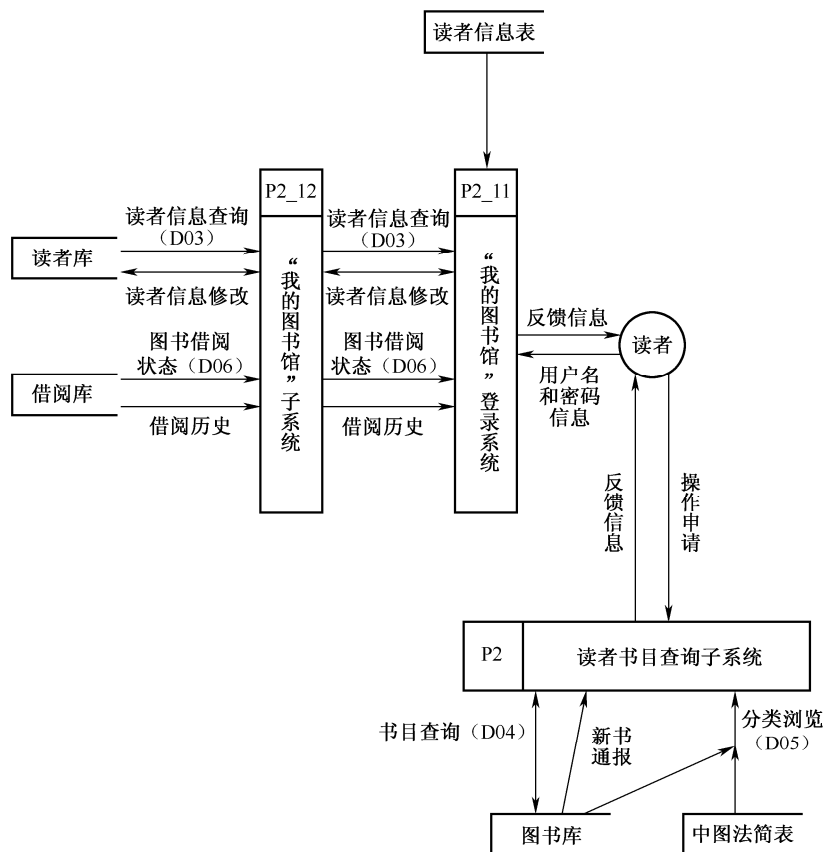


图 3-19 读者登录系统数据流程图

1.4 数据字典

图书馆书目查询系统中的部分数据字典定义如下：

(1) 数据流编号：D01。

数据流名称：图书采编信息。

简述：图书采编信息。

数据流来源：图书购买后，由图书馆采编人员编码整理后输入计算机。

数据流去向：图书采编信息将采编数据通过图书管理子系统存入图书库（图书信息表）。

数据流组成：Suoshuhao（图书索书号）+Tiaoma（图书条码号）+Timing（题名）+Zerenzhe（责任人）+Publisher（出版社）+Pubyear（出版日期）+ISBN（图书 ISBN 号）+Price（价格）+Location（馆藏地）+Zaiti（载体形式）+Buytime（购买时间）。

数据流的流量：10 本/月。

高峰流量：50 本/月。

(2) 数据流编号：D02。

数据流名称：图书借阅。

简述：图书借阅申请。

数据流来源: 读者向图书馆管理员提出书籍借阅请求, 图书馆管理员通过图书管理子系统审核后, 输入计算机。

数据流去向: 借阅库。

数据流组成: Cert_id (读者证件号)+Tiaoma (图书条码号)+Lend_date (借阅日期)+Renew_date (续借日期)+Asback_date (应还日期)+Location (馆藏地)+Timing (图书题名)+Zenrenzhe (责任者)。

数据流的流通量: 500 本/日。

高峰流量: 1000 本/日。

(3) 数据流编号: D03。

数据流名称: 读者信息查询。

简述: 返回给读者的查询结果。

数据流来源: 读者书目查询子系统, 从读者库中返给读者的查询结果。

数据流去向: 读者。

数据流组成: Cert_id (读者证件号)+Password (密码)+Name (姓名)+Sex (性别)+Id_card (身份证号)+Birthday (出生日期)+Dept (读者单位)+Authority (权限)+Cellphone (手机号码)+Tele (电话号码)+Address (联系地址)+E-mail (E-mail)+Redr_reg_day (注册日期)。

数据流的流通量: 200 次/日。

高峰流量: 500 次/日。

(4) 数据流编号: D04。

数据流名称: 书目查询。

简述: 书目查询信息。

数据流来源: 读者。

数据流去向: 读者书目查询子系统。

数据流组成: Suoshuhao (图书索书号)+Tiaoma (图书条码号)+Timing (题名)+Zerenzhe (责任者)+Publisher (出版社)+Pubyear (出版日期)+ISBN (图书 ISBN 号)+Price (价格)+Location (馆藏地)+Zaiti (载体形式)+Buytime (购买时间)。

数据流的流通量: 800 次/日。

高峰流量: 2000 次/日。

(5) 数据流编号: D05。

数据流名称: 分类浏览。

简述: 返回给读者的查询结果。

数据流来源: 读者书目查询子系统, 从图书库和中图法简表中返给读者的查询结果。

数据流去向: 读者。

数据流组成: Cname (分类名称)+Cgrade (分类等级)+Suoshuhao (图书索书号)+Tiaoma (图书条码号)+Timing (题名)+Zerenzhe (责任者)+Publisher (出版社)+Pubyear (出版日期)+ISBN (图书 ISBN 号)+Price (价格)+Location (馆藏地)+Zaiti (载体形式)。

数据流的流通量: 500 次/日。

高峰流量: 1000 次/日。

(6) 数据流编号: D06。

数据流名称: 图书借阅状态。

简述：查询图书馆中读者借阅图书数量。

数据流来源：借阅库。

数据流去向：读者。

数据流组成：Cert_id(读者证件号)+Tiaoma(图书条码号)+Lend_date(借阅日期)+Renew_date(续借日期)+Asback_date(应还日期)+Location(馆藏地)+Timing(图书题名)+Zenrenzhe(责任人)。

数据流的流量：1000 人/日。

高峰流量：2000 人/日。

1.5 开发及运行环境设置

系统开发平台：Microsoft Visual Studio 2008

系统开发语言：VB.NET

系统后台数据库：Microsoft SQL Server 2005

运行平台：Windows XP(SP3)/Windows 2000(SP4)/Windows 2003(SP2)

运行环境：Microsoft.NET Framework SDK 2.0

分辨率：最佳效果为 1024×768 像素



本章小结

(1) 需求分析就是分析软件用户的需求“是什么”，回答所要开发的应用系统将要“做什么”。通过对所要开发的目标系统的功能和性能进行详细的分析，用科学的方法来表述所要开发系统的逻辑方案，建立系统的逻辑模型，从而设计出一个合理的优化系统，确定系统的开发方向。

(2) 需求分析的主要任务就是解决所要开发的应用系统将要“做什么”的问题。需求分析就是要全面地掌握用户的各项要求，准确地表达出用户的实际需求，从而确定系统必须实现的功能。

(3) 需求分析的过程：通常把整个软件需求工程划分为需求开发和需求管理两个部分，需求分析阶段的工作可以分为问题获取、分析、编写规格说明、验证 4 个方面。

(4) 需求管理的活动：

- 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项目。
- 以一种可控制的方式将需求变更融入到项目中。
- 使当前的项目计划与需求一致。
- 估计变更需求所产生影响并在此基础上协商新的承诺，这种承诺具体体现在项目解决方案上。
- 让每项需求都能与其对应的设计、源代码和测试用例联系起来以实现跟踪。
- 在整个项目过程中跟踪需求状态及其变更情况。

(5) 软件客户需求权利书列出了 9 条关于客户在项目需求工程实施中与分析人员、开发人员交流时的合法要求，每一项权利都对应着软件开发人员、需求分析人员的义务。而软件客户需求义务书也列出了 10 条关于客户在需求过程中应承担的义务。

(6) 在软件开发过程中需求风险值得注意：

- 没有足够的用户参与。

- 用户需求的不断增加。
- 模棱两可的需求。
- 不必要的特性。
- 忽略了用户分类。
- 不准确的需求目标。

(7) 数据流程图 (Data Flow Diagram, DFD) 是一种图形化技术, 它描绘信息流和数据流从输入口转到出口的过程中所经历的变换, 既提供了功能建模机制, 也提供了信息建模机制。

(8) 软件系统中的数据流程图: 一个完整的软件系统包括系统的外部实体、数据处理、数据存储和系统中的数据流 4 个组成部分。

(9) 数据流程图的画法要点是先把整个系统按总的功能画出最粗的数据流程图, 暂不考虑系统内部的各种信息处理、传递、存储等, 只表示出系统的总体功能、系统的边界、与外界的接口等。

(10) 设计数据流程图的主要作用:

- 便于用户表达功能需求、数据需求及其联系。
- 便于两类人员共同理解现行系统和规划系统的框架。
- 清晰表达数据流的情况。
- 有利于系统建模。

(11) 分层数据流图: “分层”技术就是对数据流程图逐步深入分析, 加入各项细节。概括地说, 就是自外向内、自顶向下、逐层细化、完善求精。

(12) 根据层次关系一般将数据流程图分为顶层数据流程图、中间数据流程图和底层数据流程图。

(13) 数据字典 (Data Dictionary, DD) 就是对数据流程图中的所有被命名的组成元素作为条目加以定义, 使得每一个组成元素的名字都有一个确切的解释。所有的条目按一定次序排列, 构成一本数据字典, 提供给开发人员和用户查阅。

(14) 数据字典中有 4 类条目: 数据流、数据存储、数据元素和数据处理过程。

(15) 需求规格说明书是基于软件合同或立项建议书以及分析人员对用户现场的调研, 经过分析协商, 生成的最终相关需求文档。

(16) 需求规格说明书的内容:

- 概述。
- 系统功能。
- 数据字典。
- 细节说明。
- 数据量估计。
- 数学模型及其说明。
- 运行环境规定。

(17) 需求规格说明书的作用:

- 作为用户和软件开发商之间的合同, 为双方建立一个系统需求的文字化说明。
- 反映问题的层次和结构, 为系统设计和编码阶段提供参考依据。
- 作为软件测试、系统验收以及实施的依据。

**习题3**

1. 简述需求分析的概念？
2. 概述需求分析的主要任务。
3. 概述需求分析的过程。
4. 简述数据流程图的概念。
5. 软件系统中，数据流程图的组成部分有哪些？
6. 分层数据流程图是什么含义？
7. 简述数据字典的概念。
8. 数据字典包括哪4类条目？
9. 概述需求规格说明书的内容。
10. 概述需求规格说明书的作用。

**课后实训题**

针对学生所在学校的实际情况，自行设计一个关于校园学籍管理的系统模型，按照需求分析相关理论设计需求规格说明书，其中包括数据流程图和数据字典的定义。