

可行性研究

在客观世界中,并不是所有的问题都可以有明显的解决方法。在进行任何一项较大的工程时,首先要进行可行性分析和研究。实际上,有许多问题不可能在设定系统的规模内有解,还有一些问题在当时的技术条件下是无解的。因为这些工程中的问题并不是都有明显的解决办法,所以就不可能在预定的时间、费用之内解决这些问题。

如果这些问题没有行得通的解决办法,那么贸然开发这些项目就会造成时间、人力、资源和经费的巨大浪费。同样,对软件的项目开发也存在这一问题。所以,必须对开发项目进行可行性研究。

3.1 可行性研究任务与步骤

可行性研究与其他的研究不同,这个阶段不是去开发一个软件项目,也不是解决问题。而是研究这个软件项目是否值得去开发,其中的关键和技术难点是什么,问题能否得到解决,怎样达到目的等。一般情况下,软件可行性研究的目的是在尽可能短的时间内用最小的代价确定该软件项目是否值得去开发。

要解决这样的问题不是靠主观猜想确定的,而只能依靠客观的分析。一定要分析几种主要可能解的利弊,判断原定的目标是否现实,系统完成后所带来的经济效益是否值得投资。可行性研究实质上是要进行一次简化、压缩了的需求分析和设计过程,是要在较高层次上以较抽象的方式进行需求分析和设计过程。

可行性研究的主要内容是对问题的定义,要初步确定问题的规模和目标,问题定义后,要导出系统的逻辑模型。然后从系统的逻辑模型出发,选择若干供选择的主要系统方案。一般应从以下四个方面研究系统方案的可行性。

(1) 技术可行性研究。根据客户提出的系统功能、性能及实现系统的各项约束条件,从技术的角度研究实现系统的可行性。技术可行性研究往往是系统开发过程中难度最大的工作。由于系统分析和定义过程与系统技术可行性评估过程同时进行,这时,系统目标、功能和

性能的不确定性会给技术可行性论证带来许多困难。技术可行性研究包括风险分析、资源分析和技术分析。风险分析的任务是,在给定的约束条件下,判断能否设计并实现系统所需功能和性能。资源分析的任务是,论证是否具备系统开发所需的各类人员(管理人员和各类专业技术人员)、软件、硬件资源和工作环境等。技术分析的任务是,当前的科学技术是否支持系统一切的全过程。

(2) 经济可行性研究。进行成本效益分析,评估项目的开发成本,估算开发成本是否会超过项目预期的全部利润。分析系统开发对其他产品或利润的影响。

(3) 法律可行性研究。研究在系统开发过程中可能涉及的各种合同、侵权、责任以及各种与法律相抵触的问题。

(4) 开发方案的选择性研究。提出并评价实现系统的各种开发方案,从中选出一种用于软件项目开发。

技术可行性评估是系统可行性研究工作的关键。这一阶段决策的失误将会给开发工作带来灾难性的影响。可行性研究能保证系统开发一定明显的经济效益和较低的技术风险,一定没有各种法律问题以及其他合理的系统开发方案。如果上述四个方面中的任何一个存在问题,都应该做进一步的研究。

此外,还要为每一个可行的系统方案制定一个粗略的实现进度。可行性的根本任务是对项目的方针提出建议。可行性研究的时间依项目的规模而定,可行性研究的成本应占项目预算总成本的5%~10%。

3.1.1 研究任务

在进行项目可行性研究中,首先需要进行概要的分析研究,初步确定项目的规模和目标,确定项目的约束和限制,把它们清楚地列举出来。要研究目前正在使用的系统。如果目前有一个系统正在使用,那么这个系统一定能完成某些有用的工作。所以新系统的目标也必须能完成这些基本功能。如果现有的系统是完美的,那么用户就不会提出开发新系统。

因此,现行系统必然存在问题,也是新系统必然要解决的问题。现行系统的运行费用是一个重要的经济指标,如果新系统在相对于旧系统不能增加收入或减少开销,那么新系统就不如旧系统。研究中要阅读现行系统的资料,要实地考察。要从相关人员中了解这个系统可以做什么,为什么这样做。

一般情况下,了解到的是问题的表现,而不是实际的问题。如果要分析所得到的信息,就要从现行系统出发,该系统一定是在实际中运行的,与其他系统都有一定的联系。了解这个系统与其他系统的接口是设计新系统的重要约束条件。

可行性分析中错误的做法之一是花大量的时间去分析旧系统,了解旧系统做什么,而不是了解怎么做。第二种错误做法是不认真搜集资料,凭空想象。

从现有的物理系统出发,对物理系统进行简要的需求分析,抽象出该项目的逻辑结构,参考物理系统建立逻辑模型。

新系统的逻辑模型实质上表明系统分析员对新系统的看法。但用户是否认同了?因此系统分析员应该与用户一起对新系统进行复查问题的定义、工程的规模、目标。讨论中应该将数据流图和数据字典作为讨论的载体。如果系统分析员对于问题有误解或用户对于问题有遗漏,此时就可以改正。可行性研究的几个步骤:分析员定义问题,分析这个问题,导出

试探解,在此基础上再定义问题,再分析这个问题,修改这个问题。继续这个循环过程,直到逻辑模型完全符合系统目标。

澄清问题的定义后,系统管理人员要导出系统的逻辑模型。然后从逻辑模型出发,探索出若干种可供选择的主要解决办法,对每种解决方法都要仔细研究它的可行性。一般来说,应该从以下四个方面分析研究每种解决方法的可行性。

1. 技术可行性

要确定使用现有的技术是否能够实现系统,那么就要对开发项目的功能、性能和限制条件进行分析,确定在现有的资源条件下,技术风险有多大,项目是否能实现,这些是技术可行性研究的内容。这里的资源条件是指已有的或可以弄到的硬件、软件资源,现有技术人员的技术水平和已有的工作基础。

在技术可行性研究过程中,系统分析员应采集系统性能、可靠性、可维护性和可生产性方面的信息;分析实现系统功能和性能所需要的各种设备、技术、方法和过程;分析项目开发在技术方面可能担负的风险以及技术问题对开发成本的影响等。如有可能,应充分研究现有类似系统的功能与性能,采用的技术、工具、设备和开发过程中成功和失败的经验、教训,以便为现行系统开发作参考。必要时技术分析还包括某些研究和设计活动。

数学建模、原型建造和模拟是基于计算机系统技术分析活动的有效工具,描述了技术分析建模过程的信息流图。系统分析员通过对现实世界的观察和分析建立技术分析模型,评估模型的行为并将它们与现实世界对比,论证系统开发在技术上的可行性和优越性。基于计算机系统模型必须具备下列特性。

(1) 能够反映系统配置的动态特性,容易理解 and 操作,能够提供系统真实的结果并有利于评审。

(2) 能够综合与系统有关的全部因素,能够再现系统运行的结果。

(3) 能够突出与系统有关的重要因素,能够忽略与系统无关的或次要的因素。

(4) 结构简单,容易实现,容易修改。

技术可行性是关键的。但是,由于系统处于最初研究阶段,所以这个时候对项目的目标、功能和性能比较模糊。正因为如此,许多问题常常是最难解决的。技术可行性一般要考虑以下情况。

(1) 技术。

通过调查了解当前最先进的技术,分析相关技术的发展是否支持这个系统。

(2) 资源的有效性。

用于建立系统的硬件设备、软件、开发环境等资源是否具备?特别是用于开发项目的人员在技术和时间上是否存在问题?

如果模型很大很复杂,那么需要对模型进行分解,将一个大模型分解为若干个小模型,一个小模型的输出作为另一个小模型的输入。必要时,还可以借助模型对系统中的某一独立要素进行单独评审。开发一个成功的模型需要用户、系统开发人员和管理人员的共同努力,需要对模型进行一系列的试验、评审和修改。

根据技术分析的结果,项目管理人员必须做出是否进行系统开发的决定。如果系统开发技术风险很大,或模型演示表明当前采用的技术和方法不能实现系统预期的功能和性能,或系统的实现不支持各子系统的集成等,项目管理人员不得不做出“停止”系统开发的决定。

在评估技术可行性时,需要了解应用于本项目目前最先进的技术水平。要有相当丰富的系统开发经验,不要为了获取项目而忽略不可行的因素,对问题的评估要准确。一旦估计错误,将会出现灾难性后果。

2. 经济可行性

计算机技术发展异常迅速的根本原因在于计算机的应用促进了社会经济的发展,给社会带来了巨大的经济效益。因此,基于计算机系统的成本-效益分析是可行性研究的重要内容,它用于评估基于计算机系统的经济合理性,给出系统开发的成本论证,并将估算的成本与预期的利润进行对比。

经济可行性问题包含两方面:一方面是经济实力;另一方面是经济效益。分析经济可行性研究的内容是要进行开发成本的估算,了解项目成功取得效益的评估,确定要开发的项目是否值得投资开发。

由于项目开发成本受项目的特性、规模等多种因素的制约,对软件设计的反复优化可以获得用户更为满意的质量等,所以系统分析员很难直接估算基于计算机系统的成本和利润,得到完全精确的成本-效益分析结果是十分困难的。

一般说来,基于计算机系统的成本由以下 4 部分组成。

- (1) 购置并安装软硬件及有关设备的费用。
- (2) 系统开发费用。
- (3) 系统安装、运行和维护费用。
- (4) 人员培训费用。

在系统分析和设计阶段只能得到上述费用的预算,即估算成本。在系统一切完毕并交付用户运行后,上述费用的统计结果就是实际成本。

系统效益包括经济效益和社会效益两部分。经济效益指应用系统为用户增加的收入,它可以通过直接的或统计的方法估算;社会效益只能用定性的方法估算。

例如,开发计算机辅助设计(CAD)系统取代当前的手工设计过程。系统分析员为当前的手工设计系统和 CAD 目标系统定义对应的可测试特征,具体内容如下。

T : 绘一幅图的平均时间,单位是小时。

d : 每小时绘图的平均成本,单位是元。

n : 每年绘图的数目。

r : 用 CAD 系统绘图减少的绘图时间比例。

p : 用 CAD 系统绘图的百分比。

于是,可用下式计算利用 CAD 系统绘图每年可以节省的经费

$$B = r \times T \times n \times d \times p$$

当 $r=1/4$, $T=4$ 小时, $n=8000$ /年, $d=20$ 元/小时, $P=60\%$ 时,代入上式计算得 $B=96\,000$ 元/年,即用 CAD 系统绘图比用手工绘图平均每年约节省 96 000 元。实际上,投资利润还应该考虑软硬件降价、税收的影响和其他潜在的因素。

对于一个系统而言,一般衡量经济上是否合算,应考虑一个最小利润值。经济可行性研究范围较广,包括成本-效益分析、企业经营策略、开发所需的成本和资源、潜在的市场前景等。

3. 社会可行性

社会可行性研究的内容包括：研究开发的项目是否存在任何侵犯、妨碍等责任问题。社会可行性所涉及的范围也比较广，它包括合同、责任、侵权、用户组织的管理模式及规范，以及其他一些技术人员常常不了解的陷阱等。

有一些项目是社会公益性的。这样的项目主要是考虑在经济条件许可下的社会效益。

4. 操作的可行性

要开发项目的运行方式在用户组织内是否行得通，现有管理制度、人员素质和操作方式是否可行。

3.1.2 研究步骤

在可行性研究时，一般情况是分析人员对于项目的接触时间不长，也是分析人员对于一个新的领域技术开始认识。

同时，又要在限定条件下给出结论。因此，要求系统分析员要根据当前的技术水平和过去的经验，按照下列研究步骤执行。

1. 系统定义

系统定义是一个系统的关键，如果系统没有定义好，也就是没有确定系统的边界，就谈不上确定项目规模和目标。

为了定义好一个系统，分析员对有关人员进行调查访问，仔细阅读和分析有关材料，对项目的规模和目标进行定义和确认，描述项目的一切限制和约束，确保分析员正在解决的问题确实是要解决的问题。

2. 对于现行系统进行分析研究

现行系统可能是一个人工操作的系统，也可能是计算机上运行的旧软件系统。因某种原因需要开发一个新的计算机系统来代替现行系统。要认识到现行系统是信息的重要来源。需要研究它的基本功能、性能、环境，存在的问题，运行现行系统需要多少费用，对新系统有什么新的功能要求，新系统运行时能否减少使用费用等。

具体方法可以实地考察现行系统，收集、研究和分析现行系统的文档资料。在考察的基础上，访问有关人员，描绘现行系统的高层系统流程图，系统流程图反映了现行系统的基本功能和处理流程。还要与有关人员一起审查该系统流程图是否正确。

3. 导出新系统的逻辑模型

根据对现行系统的分析研究，搞清了新旧系统的特征，逐渐明确新系统的功能、处理流程以及所受的约束。有了这些理解后，就可以用建立逻辑模型的工具——数据流图和数据字典来描述数据在系统中的流动和处理情况。在描述时不必详细描述，只需概括地描述高层的数据处理和流动。

4. 设计方案

分析员根据新系统的高层逻辑模型，从技术角度出发，根据用户的要求和开发的技术力量，提出实现高层逻辑模型的不同方案。在设计方案时，可以设计出投资比较大的最先进的方案，可以设计出一般投资的实用方案。也可以设计出投资比较小的基本方案，提供以后进

行比较,然后再根据技术可行性、经济可行性和社会可行性对各种方案进行评估,去掉行不通的解法,就得到了可行的解法。

5. 推荐可行的方案

根据上述可行性研究的结果,同时根据用户的具体情况,来决定该项目是否值得去开发。若值得开发,那么可行的解决方案是什么,并且说明该方案可行的原因和理由。该项目是否值得开发的主要因素,是从经济上看是否合算,这就要求分析员对推荐的可行方案进行成本-效益分析。

6. 编写可行性研究报告

将上述可行性研究过程的结果按照:说明要求、目的、条件与限制、可行性研究方法及评价尺度;处理流程、工作负荷、费用开销和局限性;说明处理流程、运行环境和局限性;技术条件的可行性;经济方面的可行性;社会条件的可行性;其他可供选择的系统;结论的顺序写成可行性研究报告,提请用户和使用部门仔细审查,从而决定该项目是否进行开发,是否接受可行的实现方案。

3.2 系统分析

如果确认开发一个新的软件系统是必要而且可能的,那么就要进入系统分析阶段。这个时期的首要任务是认识和对问题的评价、建立模型和对规格的分析。要清楚软件工作域是进行各种估算的基础。为了了解软件在系统中的各种关系和评审软件工作域。系统分析员要研究系统规格说明(System Specification)和软件项目计划(Software Project Plan)。其次,为了确保对问题的识别,必须为分析建立通信关系。系统分析员必须与用户和软件开发机构的管理与技术人员进行接触。项目管理员可以作为协调员来保证通信渠道的畅通。系统分析员的目标是弄清用户已经理解的基本问题元素。

第二项任务是分析,是主要工作问题评价与解的综合。系统分析员必须定义和详细描述全部软件功能,熟悉影响系统事件前后关系的软件行为,建立系统界面的特征,评价信息流和信息的内容,以及揭示设计限制。这些任务都是为问题的描述服务的,所以整个方法或解是一个综合的过程。

最后一项任务是需求分析文档(规格说明和用户手册)。它是作为用户和开发人员进行评审的基础。一般情况下,需求评审经常导致软件功能、性能、信息表达式、限制和确认准则的修改。由于软件的特性所至,如果没有分析文档,在进行评审的过程中,可能因为没有这种基础而不能发现运行中对软件功能、性能、信息表达式、限制和确认准则所存在的问题。此外,软件项目计划也是由此得到评价,它可以决定早期的各种估算是否仍然有效。

3.2.1 系统分析员

要完成上述系统分析的任务,就要有相应的技术人员,即系统分析员。

对系统分析员职务有许多不同叫法:系统分析员、主系统设计员、分析员等。不管其职务叫什么,作为系统分析员必须具备下列能力:

(1) 能掌握抽象概念(Abstract Concepts),并能将其整理为逻辑划分(Logical Divisions),

以及根据每一个逻辑划分综合为解(Solutions)的能力。

- (2) 有弄清用户环境的能力。
- (3) 有从冲突(Conflict)或混淆(Confusions)中吸取恰当事实的能力。
- (4) 有用较好的书面和口头形式进行通信(Communication)的能力。
- (5) 有把硬件和软件系统用于用户环境(User/Customer Environments)的能力。
- (6) “从树木见森林”的能力。

上述所列的最后一条也许就是区分一位杰出的系统分析员与一般的分析者的标准。由过去的经验可知,一般分析者往往在项目一开始就考虑一些细小的环境和实现,过早地拘泥于细节,而忽略整个软件目标。系统分析员要完成或合作完成与软件需求分析有关的每一项任务。在整理各项任务过程中,他应与用户进行交流,以确定现有环境的特征。系统分析员在评价和综合任务中,也要这样要求开发人员。这样,软件的特性才能被正确定义。系统分析员的一般职责是负责软件需求规格说明的开发,并参与整个评审。

系统分析员必须弄清每一个软件工程模式,并懂得其中的每一步。有许多隐含的软件需求(如可维护性设计),只有懂得软件工程的分析者,才能把它包含在需求规格之中。

需求分析是软件工程的一个重要任务,它是系统层软件配置与软件设计之间的桥梁。系统分析员在软件需求分析阶段有五个方面的工作:

- (1) 问题识别(Problem Recognition)。
- (2) 评价和综合(Evaluation and Synthesis)。
- (3) 建模(Modeling)。
- (4) 规格说明(Specification)。
- (5) 评审(Review)。

3.2.2 面临的问题域

在系统分析的过程中,都会遇到许多问题,特别是在 3.2.1 节讨论的前两项任务。问题识别与问题评价和解综合,在很大程度上决定于能否获得恰当的信息。一般情况下,会有用户提供的信息与早期其他人提出的需求说明相矛盾;也有功能和性能与其他系统元素所给的限制相矛盾;特别是随着时间的推移,对系统目标的理解发生变化。那么,应当收集什么信息?应当怎样对它们进行表示?谁能提供各种信息的初始模型?以及采用什么技术和工具才能方便地进行信息收集?所有这些都要解决的问题。

随着问题规模的增大,分析任务的复杂性也在增加。每个新的信息项、功能或限制都可能影响整个软件的其他元素。由此,随着问题复杂性增加,分析工作量将呈几何级数增长。问题就是当定义一个大系统时,怎样能避免系统内部的不一致性?怎样能察觉这种遗漏?为了更易于处理大问题怎样能有效地进行划分?等等。

需求分析是一项要求充分通信的活动过程。通信时,通信中的误解和遗漏都可能给系统分析员和用户造成困难。需求分析中遇到的难题,必须要与已得到的恰当信息、处理问题的复杂性,以及在分析以后可能发生的相应变化联系起来。

系统工程第一定律就是无论在系统生存期的哪个阶段,系统都将发生改变,而且这种改变要求将在整个生存期中连续不断。这里所讲的变化是需求的变化。不论讲的是一个系统或只是软件,改变都是要发生的。事实上,分析任务完成的前后,这种改变的要求肯定不断

出现。问题是软件需求怎样与其他系统元素改变相适应?怎样评价一个改变可能对其看起来无关的软件部分的影响?以及为了避免副作用,在规格说明中怎样进行错误的改正?

造成上述问题有许多原因,可归纳如下:

- (1) 缺少通信,使信息获得困难。
- (2) 由于不适当的技术和工具致使规格说明不充分或不准确。
- (3) 在需求分析中,试图走捷径,导致不可靠的设计。
- (4) 在软件定义前,方案选择错误。

上述这些问题及其产生的原因,不可能用需求分析的软件工程方法来彻底解决。但是固定通信技术的应用、基本分析原理,以及系统的分析方法都将极大地减少上述问题的影响。

3.2.3 通信技术

一个软件开发的开始往往是用户提出一个问题,并认为这个问题可能适合用计算机来解决,于是寻找开发者,这时开发者对用户的请求回答是可以对用户进行帮助。这样,开发者与用户间的通信就开始了。所以说软件需求分析通常是从两方面或多方面之间的通信开始的。但是,从通信到真正理解系统常常是艰难的。

1. 过程的开始

当用户与开发者有了合作意向后,经常使用的分析技术是会议或访问,将它作为用户和开发人员之间的通信桥梁。会议或访问期间,双方都希望能成功。对于系统总的目标有一个模糊的想法,对于具体的目标和要求常常不知道应该问什么,说什么,一般情况下在第一次的会谈中用户会介绍自己的基本情况。建议:系统分析员可以就此切入,了解用户的现行运作,希望在哪些地方用计算机来解决问题,提出用户、总目标和效益方面的问题,如下:

- (1) 这项工作主要在哪些部门应用?
- (2) 项目的结构主要是谁使用?
- (3) 项目成功的应用带来什么样的经济效益?
- (4) 为了达到目的,还需要其他什么资源?

在此过程中用户一般会介绍他们的现状。为了利于系统分析员对问题更好地理解,而用户又能够表达关于目标的理解。可以进一步了解以下内容。

- (1) 如何表示一个成功的目标能产生一个好的输出形式?
- (2) 这样的理解可以解决一些什么样的问题?
- (3) 显示或描述一下这样的目标要使用的环境。
- (4) 这种方式的解对特殊的性能问题或限制将有哪些影响?

通过初步了解后,系统分析员会发现一些更具体和技术上的问题。为了通信的有效性,要考虑的称为元问题(Meta-Questions)。建议如下:

- (1) 谁是能回答这些问题的人?
- (2) 谁能贴切解决这些问题?
- (3) 问题是否提得太多?
- (4) 这里还有其他人可以提供其他方面的信息吗?

(5) 还有什么事情是应该问的?

上述所有这些问题将有助于打开僵局和初步的通信,这种通信是成功分析的基础。在交流的过程中最好不要用一问一答的会议形式。事实上,问答式会议应当只用于第一次会面,然后采用交换式会议的形式,综合问题的各个元素协商和说明。这种类型的会议方法在下面还要讨论。

2. 深入了解

在交流的过程中,用户因为对自己的业务非常熟悉,所以会用很简短的概括叙述。这时可以借此向用户索取有关业务源头与结果的资料,通过对资料的分析,对于一些复杂的业务,要到现场去跟随业务流程“跟单”,然后把理解的过程用合适的技术表示出来。这种业务过程的表示是否正确,要与用户共同分析和确认。

3. 方便的应用规范技术

用户和软件工程师通常有一种无意识的“我们和你们”的思想,而不是作为一个组的工作来确定和定义需求。每个用户用他们自己的方式来定义,或通过备忘录、文件、问答式会议的通信方式来定义。历史表明,这种方法会使许多误解和重要信息被遗漏,而且建立起来的绝不是一种成功的工作关系。

由于在思想上存在“你我”的问题,所以一些研究机构开发出一种面向组的方法,应用于分析和规格说明早期的需求收集,这种方法叫做方便的应用规范技术(Facilitated Application Specification Technique, FAST)。FAST 促使用户和开发人员组成一个联合组一起去确定问题,提出解的各个元素,协商不同方法,并定义一个初步解的需求集。今天,FAST 已广泛地应用于信息系统界。这种技术对于促进通信在所有类型的应用中具有潜力。现在,FAST 有许多不同的方式,每种方式的使用都有不同的背景,但所有应用都必须遵循以下原则:

- (1) 会议由上一级部门或请中立部门主持,开发人员和用户双方参加。
- (2) 一位主持人(可以是用户、开发商或局外人),是被指定控制会议的人。
- (3) 确定准备和参加的原则。
- (4) 提出一个议事日程,这个议事日程是正规的,并包括所有重要观点。但是要不拘礼节地鼓励大家自由发表意见。
- (5) 一张确定的日程(可以是工作单、可转动的图表、墙上张贴物或墙上印刷牌)。
- (6) 目标就是确定问题,提出解的各种元素,协商不同方法,以及定义一个初步解的需求集,而且能在一个有助于目标完成的气氛中进行。

3.3 分析原理

在过去已经开发出一些软件分析和规格说明的方法,这些软件分析和规格说明的方法虽然在软件的开发中起到了积极的作用,但是应用中也发现了存在的问题及其原因。为此又开发出一些方法和规则来克服这些问题。于是研究出了多种软件分析和规格说明的方法,每一种方法都有它不同的表示法和观点。归纳起来,所有的分析方法都与下述一组基本原理相联系:

- (1) 问题的信息域必须能被表示和被理解。

- (2) 应当开发描述系统信息、功能和行为的模型。
- (3) 问题必须能按一定形式进行分割,就是用一种层次(或分层)形式来揭示它们的细节。
- (4) 分析过程应当从基本信息开始,直到实现细节。

不论用何种软件分析和规格说明的方法,都是根据这些原理的应用。有了这些基本原理,分析者可以系统地研究问题。为了更全面地了解功能检查信息域;为了按合同方式进行信息交流使用模型;为了降低问题的复杂性采用分割技术;而软件的基本和实现观点,就是软件需要符合由于处理需求所造成的逻辑限制,同时还要符合由于其他系统元素所造成的物理限制。

3.3.1 信息域

在计算机的应用领域中,所有的软件应用都可以被统称为数据处理(Data Processing)。有了数据处理这个概念,就非常利于了解软件需求。所谓软件就是对处理数据进行构造。这种构造表现为将数据从一种形式变换到另一种形式,即接受输入,继以进行某种方式处理,然后产生输出。信息处理也指处理代表信息的数据并确定被处理数据的意义的过程,所以可以用信息处理代替数据处理。信息处理的对象是信息,而信息的定义是:一方面是物质状态发生改变的一种表征,通常指数据消息中所包含的意义;另一方面它是知识的一种元素,以任何形式聚合,能产生一完整的概念、条件或情况的数据,通信中线路传输时被加入接收消息内的数据。

信息域是一信息字或一组信息中的特定部分,信息域中的内容通常被作为一个整体来处理。它包括三种不同方面的数据和控制:信息内容、信息结构和信息流。每个方面都由计算机程序处理。

在需求分析的过程中,搜集与分析的主要对象是信息域,为了完全地了解信息域,对信息的每一个方面都应该进行研究。

1. 信息内容

知识的一种元素,描述了单个数据和控制项,这些项可以组成更大的信息项。

2. 信息结构

信息结构描述了各种数据和控制项的内部组织。数据和控制项组成的是一个 n 维的表还是一个分层的树结构? 在结构内部,哪些信息与其他信息相关? 所有的信息都包含在一个单一结构内,还是使用了不同的结构? 在一个信息结构内怎样与另外的结构信息相关联? 有关软件信息结构设计与实现的分析讨论,由于与数据结构的概念相关,因此在后面讨论。

3. 信息流

信息流描述了数据和控制沿系统流动变化的方式。信息流可以用数据流图(Data Flow Diagram, DFD)来表示,如图 3-1 所示。

从图 3-1 可以看出,输入信息经变换后为中间信息,然后再变换为输出信息。沿这条变换路径,可以从存在的数据库(磁盘文件或缓冲存储器)引入另外的信息。应用于数据的这种变换必须完成的程序功能或子功能,在两个变换(功能)之间流动的数据和控制定义了每个功能的接口。

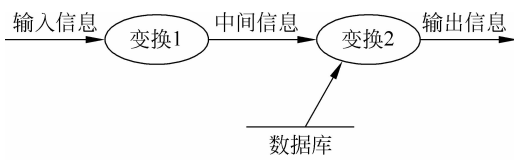


图 3-1 信息流

3.3.2 建立模型

在科学的各个领域,为了更好地理解建立的实际实体。对研究的系统、过程、事物或概念的一种表达形式,可以是物理实体,也可以是图表或数学表达式。当这种实体是一个物理的物体时,就可以用缩小比例的方法建立一种形式表示的模型。而当构造的实体是软件时,模型必须采用另一种形式。它必须能够模拟软件变换的信息,能使变换的功能(和子功能)作为变换系统的行为发生。

在软件需求分析过程中,要完成将建立的系统的模型。模型主要说明系统必须做什么,而不是表达怎样做。在一般情况下,用图形符号表示的方法来描述信息、处理、系统行为,而其他特性则使用性质不同的和公认的图符(Icons)表示。如果条件允许,可以采用源代码通过在机器上演示说明系统做什么,模型的另外一些部分可以是纯的文字形式,对需求的描述可以用自然语言或专用语言。

在需求分析中,模型的建立可以反映人们对事物的认识,起到很重要的作用。

首先,这种模型可以辅助分析人员更好地了解系统的信息、功能和行为,从而使分析更容易和更系统化。

其次,模型是评审的焦点,是确定系统完整性(Completeness)、一致性(Consistence)和规格说明准确性(Accuracy)的关键。

最后,模型也是设计的基础。模型能给设计人员提供一种软件的基本表达式,这种表达式可以映射成为实现的正文。

以后要讨论的各种分析方法,实际就是建立模型方法。对于比较小的软件系统,可以省略建立模型,但对于比较大的系统最好要建立模型。建立模型方法的作用是做好分析工作的基础。

3.3.3 分解

在相关领域中,通常都会有涉及多技术的复杂问题分析。把一个很大和复杂的问题作为一个整体很难被完全理解。因此,为了能够较容易地理解这个问题,人们力图把这样的问题分解为若干部分。与此同时,为了整体功能的完成,还要在各个部分之间建立它们的接口。在需求分析过程中,软件领域的信息、功能、行为都可以分解。

从本质上讲,分解就是把一个问题划分为几个组成部分。在概念上,可以建立一种分层的功能或信息表达式,然后按下述两条分解出最主要的元素:

- (1) 在分层中,按垂直方向逐层细化。
- (2) 在分层中,按水平方向对功能进行分解。

为了说明分解的方法,现在分析一个文字处理软件系统的例子。

对于文字处理软件的需求,可以根据产品划分的信息、功能、行为等几部分来分析。图 3-2(a)和图 3-2(b)分别给出了文字处理软件功能部分的水平分解和垂直分解。

软件需求分析应当集中在软件完成什么上,而不在怎样才能实现处理上。然而,这里不应该把实现观点作为怎样做的表达式来理解。相反,一个实现模型给出了当前的操作方式,只是把已有的和建议的操作分配给系统的所有元素。而基本模型(功能的或数据的)在一般意义上,是不应直接给出功能怎样实现的。