

第3章 需求工程

软件需求是决定软件开发是否成功的一个关键因素,一旦发生错误,将会给整个软件开发工作带来极大的损害,并给以后的软件维护带来极大的困难。整个软件工程活动中,任何一个其他阶段都不能称为工程,只有需求阶段可以称为需求工程。这体现了需求在整个软件工程过程中非同一般的重要性。需求工程的所有过程包括需求的获取、需求分析、需求定义、需求验证和需求管理。

3.1 软件需求

关于软件需求的定义,在 IEEE 软件工程标准词汇表(1997 年)中给出如下描述:

- (1) 用户解决问题或达到目标所需的条件或能力。
- (2) 系统或系统部件要满足合同、标准、规范或其他正式规定文档所需具有的条件或能力。
- (3) 一种反映上面(1)或(2)所描述的条件或能力的文档说明。

从这个标准定义中可以看出需求在软件工程中的重大作用,也可以看出需求的概念涵盖了用户角度(系统的外部行为)和开发人员角度(系统的内部特性)两个方面。

通常,软件需求可以分为不同的层次:业务需求、用户需求以及功能需求和非功能需求,它们之间的关系如图 3.1 所示。

业务需求反映了组织机构或客户对系统和产品高层次的目标要求,它们在项目视图与范围文档中予以说明。用户需求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说明中予以说明。功能需求和非功能需求定义了开发人员必须实现的软件功能,这些需求则体现在需求文档中。

3.1.1 业务需求

业务需求说明了提供给客户和产品开发商的新系统的最初利益,反映了组织或客户对系统高层次的目标要求,它们在项目视图与范围文档中予以说明。项目视图和范围文档应该包括高层的产品业务目标,所有的使用实例和功能需求都必须能达到业务需求的需要。项目视图说明使所有项目参与者对项目的目标能达成共识。而范围则是作为评估需求或潜在特性的参考。

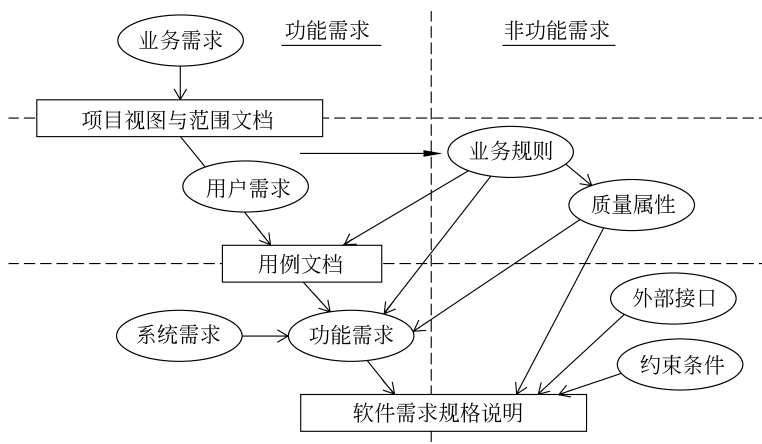


图 3.1 不同层次的软件需求及其关系

3.1.2 用户需求

用户需求是从用户角度描述的系统功能需求和非功能需求,通常只涉及系统的外部行为,而不涉及系统的内部特性。用户需求文档描述了用户使用软件产品要完成的任务,用户需求描述的原则是:应该易于用户的理解,一般不采用技术性很强的语言,而是采用自然语言和直观图形相结合的方式进行描述。

3.1.3 功能需求

功能需求描述系统所预期提供的功能或服务。即定义系统应该做什么,系统要求输入什么信息,输出什么信息,以及如何将输入变换为输出。它由开发的软件类型、软件未来的用户以及开发的系统类型决定。

3.1.4 非功能需求

除了关心软件的功能和行为外,用户会将更多注意力集中于诸如产品的易用性、运行速度、可靠性、如何进行异常处理等特性。因此,一个好的软件还必须满足一系列非功能需求。非功能需求是指那些不直接与系统具体工作相关的一类需求。主要涉及系统的总体特性,如可靠性、反应时间和储存空间等。

非功能需求涉及的面非常广,包括系统的性能,目标系统所受的限制条件和开发与维护软件的限制,还包括如安全规章、隐私权保护的立法等外部因素。具体内容可由三个方面构成:产品需求、机构需求、外部需求,细节如图 3.2 所示。

对于非功能需求的表述要求尽可能地量化且可验证。表 3.1 给出了一些可能用来指定非功能性系统特性的度量,据此可以检验系统是否满足了相应的需求。

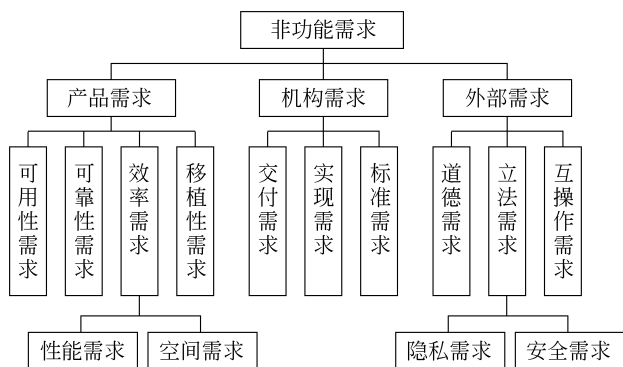


图 3.2 非功能需求

表 3.1 指定非功能需求的度量

特性	度量指标
速度	每秒处理的事务 用户/事件响应时间 屏幕刷新时间
规模	K 字节 RAM 芯片数
易用性	培训时间 帮助画面数
可靠性	失败平均时间 无效的概率 失败发生率 有效性
鲁棒性	失败之后的重启次数 事件引起失败的百分比 失败中数据崩溃的可能性
可移植性	依赖于目标的语句百分比 目标系统数

3.2 需求工程过程

需求工程是随着计算机软件的发展而发展起来的。所谓软件需求工程是一门应用有效的技术和方法、合适的工具和符号来确定、管理和描述目标系统及其外部行为特征的学科。在需求工程过程中,开发人员需要收集和分析来自用户或市场等各方面的需求,编写规格说明文档,并采用评审和商议等有效手段对其进行验证,最终形成一个需求基线。由于软件开发过程中经常发生需求变更的情况,因此必须有效地管理和控制这些变更。

因此需求工程过程包括需求的获取、需求分析、需求定义、需求验证和需求管理。

1. 需求获取

确定和收集与待开发的软件系统相关的用户需求信息。

2. 需求分析

对获得的用户需求信息进行分析和综合,找出错误和冲突及遗漏的地方,获得用户的准确需求,进而建立软件系统的逻辑模型或需求模型。

3. 需求定义

利用描述语言、标准格式书写软件系统的需求规格说明。

4. 需求验证

审查和验证软件系统需求规格说明,进而确定需求规格说明是否正确描述了用户对软件系统的需求。

5. 需求管理

需求管理的任务是:管理软件系统的需求规格说明,评估需求变更带来的影响及成本费用,跟踪软件需求的状态,管理需求规格说明的版本等。

下面分别叙述需求工程的每一部分。

3.3 需求的获取

需求获取是在问题及其最终解决方案之间架设桥梁的第一步。需求获取的关键在于通过与用户的沟通和交流,收集和理解用户的各项要求。

3.3.1 需求获取的过程

需求获取的主要任务是与客户或用户沟通,了解系统或产品的目标是什么,客户或用户想要实现什么?系统和产品如何满足业务的要求,最终系统或产品如何用于日常工作,等等。获取并理解用户的需求是软件工程师所面对的最困难的任务之一。

对于不同规模及不同类型的项目,需求获取的过程不会完全一样。下面给出需求获取过程的参考步骤。

1. 开发高层的业务模型

所谓应用领域,即目标系统的应用环境,如银行、电信公司等。如果系统分析员对该领域有了充分了解,就可以建立一个业务模型,描述用户的业务过程,确定用户的初始需求。然后通过迭代,更深入地了解应用领域,之后再对业务模型进行改进。

2. 定义项目范围和高层需求

在项目开始之前,应当在所有利益相关方中建立一个共同的远景,即定义项目范围和高层需求。项目范围描述系统的边界以及和外部事物(包括组织、人、硬件设备、其他软件等)的关系。高层需求不涉及过多的细节,主要表示系统需求的概貌。

3. 识别用户类和用户代表

需求获取的主要目标是理解用户需求,因而客户的参与是生产出优质软件的关键因素。因此,首先确定目标系统的不同用户类型;然后挑选出每一类用户和其他利益相关方的代

表,并与他们一起工作;最后确定谁是项目需求的决策者。

用户类可以是人,也可以是与系统打交道的其他应用程序或硬件部件。如果是其他应用程序或硬件部件,则需要以熟悉这些系统或硬件的人员作为用户代表。

4. 获取具体的需求

确定了项目范围和高层需求,并确定了用户类及用户代表后,就需要获取更具体、完整和详细的需求。具体需求的获取方法下节详细介绍。

5. 确定目标系统的业务 workflow

具体到当前待开发的应用系统,确定系统的业务 workflow 和主要的业务规则,采取需求调研的方法获取所需的信息。例如,针对信息系统的需求调研方法如下:

(1) 调研用户的组织结构、岗位设置、职责定义,从功能上区分有多少个子系统,划分系统的大致范围,明确系统的目标。

(2) 调研每个子系统的工作流程、功能与处理规则,收集原始信息资料,用数据流来表示物流、资金流、信息流三者的关系。

(3) 对调研内容事先准备,针对不同管理层次的用户询问不同的问题,列出问题清单。将操作层、管理层、决策层的需求既联系又区分开来,形成一个需求的层次。

6. 需求整理与总结

必须对上面步骤取得的需求资料进行整理和总结,确定对软件系统的综合要求,即软件的需求。并提出这些需求实现条件,以及需求应达到的标准。这些需求包括功能需求、性能需求、环境需求、可靠性需求、安全保密要求、用户界面需求、资源使用需求、软件成本消耗与开发进度需求等。

3.3.2 需求获取的常用方法

需求获取的常用方法:

1. 用户面谈

它是一种理解商业功能和商业规则的最有效方法。面谈过程需要认真地进行计划和准备,其基本要点如下所述。

(1) 面谈之前:确立面谈目的;确定要包括的相关用户;确定参加会议的项目小组成员;建立要讨论的问题和要点列表;复查有关文档和资料;确立时间和地点;通知所有参加者有关会议的目的、时间和地点。

(2) 进行面谈时:衣着得体;准时到达;寻找异常和错误情况;深入调查细节;详细记录;指出和记录未回答条目和未解决问题。

(3) 面谈之后:复查笔记的准确性、完整性和可理解性;把所收集的信息转化为适当的模型和文档;确定需要进一步澄清的问题域;适当的时候向参加会议的每个人发一封感谢信。

2. 需求专题讨论会

需求专题讨论会也许是需求获取的一种最有力的技术。项目主要风险承担人在短暂而紧凑的时间段内集中在一起,一般为 1~2 天,与会者可以在应用需求上达成共识、对操作过程尽快取得统一意见。参加会议人员包括:主持人、用户、技术人员、项目组人员。

专题讨论会具有以下优点：

- (1) 协助建立一支高效的团队,围绕项目成功的目标。
- (2) 所有的风险承担人都畅所欲言。
- (3) 促进风险承担人和开发团队之间达成共识。
- (4) 揭露和解决那些妨碍项目成功的行政问题。
- (5) 能够很快地产生初步的系统定义。

3. 问卷调查

问卷调查可用于确认假设和收集统计倾向数据,问卷需要快速回答,允许匿名方式。存在问题的是:相关的问题不能事先决定;问题背后的假设对答案造成偏颇,如这符合你的期望吗;难以探索一些新领域;难以继续用户的模糊响应。在完成最初的面谈和分析后,问卷调查可作为一项协作技术可以收到良好的效果。

4. 现场观察

掌握用户如何实际使用一个系统以及到底用户需要哪些信息,最好的办法是亲自观察用户是如何完成实际工作的。一般方法是:

- (1) 对办公室进行快速浏览,了解布局、设备要求和使用、工作流总体情况。
- (2) 安排几个小时观察用户是如何实际完成他们的工作,理解用户实际使用计算机系统和处理事务的细节。
- (3) 像用户一样接受训练和做实际工作,发现关键问题和瓶颈。

5. 原型化方法

一个软件原型是所提出的新产品的部分实现,可以帮助开发人员、用户以及客户更好地理解系统的需求,它比开发人员常用的技术术语更易于理解。建立原型可以解决在产品开发的早期阶段需求不确定的问题,用户、经理和其他非技术项目风险承担者发现在确定和开发产品时,原型可以使他们的想象更具体化。如建立基于 Web 的应用系统原型,使用 HTML 进行界面设计。

6. 基于用例的方法

随着面向对象技术的发展,基于用例的方法在需求获取和建模方面的应用越来越普遍。用例建模是以任务和用户为中心的,开发和描述用户需要系统做什么。另外,用例有助于开发人员理解用户的业务和应用领域,并可以运用面向对象分析和设计的方法将用例转化为对象模型。

用例建模的基本步骤:

- (1) 确定系统的参与者:参与者是与系统交互的外部实体,它既可以是人员也可以是外部系统或硬件设备。
- (2) 确定场景:场景是对人们利用计算机系统过程做了什么和体验了什么的叙述性描述。它从单个参与者的角度观察系统特性的具体化和非正式的描述。
- (3) 确定系统用例:用例描述了一个完整的系统事件流程,其重点在于参与者与系统之间的交互而不是内在的系统活动,并对参与者产生有价值的可观测结果。
- (4) 确定用例之间的关系:在确定出每一个参与者的用例之后,需要将参与者和特定的用例联系起来,最终绘制出系统的用例图。
- (5) 编写用例描述文档:单纯使用用例图并不能提供用例所具有的全部信息,因此需

要使用文字描述那些不能反映在图形上的信息。用例描述是关于角色与系统如何交互的规格说明,要求清晰明确,没有二义性。在描述用例时,应该只注重外部能力,不涉及内部细节。

3.4 需求分析

3.4.1 需求分析的特点

需求分析是指开发人员要准确理解用户的要求,进行细致的调查分析,将用户非形式的需求陈述转化为完整的需求定义,再由需求定义转换到相应的形式功能规约(需求规格说明)的过程。需求分析虽处于软件开发过程的开始阶段,但它对于整个软件开发过程以及软件产品质量是至关重要的。在计算机发展的早期,所求解问题的规模较小,需求分析被忽视。随着软件系统复杂性的提高及规模的扩大,需求分析在软件开发中所处的地位愈加突出,从而也愈加困难,它的难点主要体现在以下几个方面。

(1) 问题的复杂性。这是由用户需求所涉及的因素繁多引起的,如运行环境和系统功能等。

(2) 交流障碍。需求分析涉及人员较多,如软件系统用户、问题领域专家、需求工程师和项目经理等,这些人具备不同的背景知识,处于不同的角度,扮演不同角色,造成了相互之间交流的困难。

(3) 不完备性和不一致性。由于各种原因,用户对问题的陈述往往是不完备的,其各方面的需求还可能存在着矛盾,需求分析要消除其矛盾,形成完备及一致的定义。

(4) 需求易变性。用户需求的变动是一个极为普遍的问题,即使是部分变动,也往往会影响到需求分析的全部,导致不一致性和不完备性。

为了克服上述困难,人们主要围绕着需求分析的方法及自动化工具(如 CASE 技术)等方面进行研究。

3.4.2 需求分析的原则

目前存在着许多需求分析的方法,虽然各种方法都有其独特的描述方法,但不论采用何种方法,需求分析都必须遵循以下 3 个基本原则。

1. 能够表达和理解问题的数据域和功能域

所有软件开发的最终目的都是为了解决数据处理的问题,数据处理的本质就是将一种形式的数据转换成另一种形式的数据,即通过进行一系列加工将输入的原始数据转换为所需的结果数据。需求分析阶段必须明确系统中应具备的每一个加工、加工的处理对象和由加工所引起的数据形式的变化。

2. 能够将复杂问题分解化简

为了便于问题的解决和实现,在需求分析过程中需要对于原本复杂的问题按照某种合适的方式进行分解(对功能域和数据域均可)。分解可以是同一层次上的横向分解,也可以

是多层上的纵向分解。每一步分解都是在原有基础上对系统的细化,使系统的理解和实现变得较为容易。

3. 能够给出系统的逻辑表示和物理表示

系统需求的逻辑表示用于指明系统所要达到的功能要求和需要处理的数据,不涉及实现的细节。系统需求的物理表示用于指明处理功能和数据结构的实际表现形式,通常由系统中的设备决定。如处理数据的来源,某些软件可能由终端输入,另一些软件可能由特定设备提供。给出系统的逻辑表示和物理表示对满足系统处理需求所提出的逻辑限制条件和系统中其他成分提出的物理限制是必不可少的。结构化分析方法和面向对象分析方法都遵循以上原则。

3.4.3 需求分析的任务

需求分析的基本任务是准确地回答“系统必须做什么?”这个问题。需求分析的任务不是确定系统怎样完成工作,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。需求分析阶段的具体任务如下所述。

1. 确定综合需求

对系统的综合需求有下述四个方面。

(1) 系统功能需求:应该划分出系统必须完成的所有功能。

(2) 系统性能需求:指待开发的软件的技术性能指标,如存储容量、运行时间等限制。

(3) 环境的需求:指软件运行时所需要的软、硬件(如机型、外设、操作系统和数据库管理系统等)的要求。

(4) 将来可能提出的需求:应该明确地列出那些虽然不属于当前系统开发范畴,但是据分析将来很可能会提出来的要求。这样做的目的是在设计过程中对系统将来可能的扩充和修改预做准备,以便一旦需要时能比较容易地进行这种扩充和修改。

2. 分析系统的数据要求

任何一个软件系统本质上都是信息处理系统,系统必须处理的信息和系统应该产生的信息在很大程度上决定了系统的面貌,对软件设计有深远影响,因此,必须分析系统的数据要求,这是软件需求分析的一个重要任务。分析系统的数据要求通常采用建立概念模型的方法。复杂的数据由许多基本的数据元素组成,数据结构表示数据元素之间的逻辑关系。利用数据字典可以全面准确地定义数据,但是数据字典的缺点是不够形象直观。为了提高可理解性,常常利用图形工具辅助描绘数据结构。常用的图形工具有层次方框图和Warnier图。

3. 导出系统的逻辑模型

综合上述两项分析的结果可以导出系统的详细的逻辑模型,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. 编写文档

编写文档的步骤如下:

(1) 编写“需求规格说明书”,把双方共同的理解与分析结果用规范的方式描述出来,作为今后各项工作的基础。

(2) 编写初步用户使用手册,着重反映被开发软件的用户功能界面和用户使用的具体要求,用户手册能强制分析人员从用户使用的观点考虑软件。

(3) 编写确认测试计划,作为今后确认和验收的依据。

(4) 修改完善项目开发计划。在需求分析阶段对开发的系统有了更进一步的了解,所以能更准确地估计开发成本、进度及资源要求,因此对原计划要进行适当修正。

3.4.4 需求分析的方法

需求分析方法有功能分解方法、结构化分析方法、信息建模方法和面向对象分析方法等。

1. 功能分解方法

功能分解方法是将一个系统看成是由若干功能构成的一个集合,每个功能又可划分成若干个加工(即子功能),一个加工又进一步分解成若干加工步骤(即子加工)。这样,功能分解方法由功能、子功能和功能接口三个组成要素。它的关键策略是利用已有的经验,对一个新系统预先设定加工和加工步骤,着眼点放在这个新系统需要进行什么样的加工上。

功能分解方法本质上是用过程抽象的观点来看待系统需求,是符合传统程序设计人员的思维特征,而且分解的结果一般已经是系统程序结构的一个雏形,实际上它已经很难与软件设计明确分离。

这种方法存在一些问题,它需要人工来完成从问题空间到功能和子功能的映射,即没有显式地将问题空间表现出来,也无法对表现的准确程度进行验证,而问题空间中的一些重要细节更是无法显示出来。功能分解方法缺乏对客观世界中相对稳定的实体结构进行描述,而基点放在相对不稳定的实体行为上,因此,基点是不稳定的,难以适应需求的变化。

2. 结构化分析方法

结构化分析方法是一种从问题空间到某种表示的映射方法,由数据流图表示软件的功能,是结构化方法中重要的、被普遍接受的表示系统,它由数据流图和数据词典构成。这种方法简单实用,适于数据处理领域问题。

该方法对现实世界中的数据流进行分析,把数据流映射到分析结果中。但现实世界中的有些要求不是以数据流为主干的,就难于用此方法。如果分析是在现有系统的基础上进行的,应先除去原来物理上的特性,增加新的逻辑要求,再追加新的物理上的考虑。这时,分析面对的并不是问题空间本身,而是过去对问题空间的某一映射,在这种焦点已经错位的前提下,来进行分析显然是十分困难的。

该方法的一个难点是确定数据流之间的变换,而且数据词典的规模也是一个问题,它会引起所谓的“数据词典爆炸”,同时对数据结构的强调很少。

3. 信息建模方法

信息建模方法是从数据的角度来对现实世界建立模型的,它对问题空间的认识是很有帮助的。

该方法的基本工具是 E-R 图,其基本要素由实体、属性和联系构成。该方法的基本策略是从现实世界中找出实体,然后再用属性来描述这些实体。

信息模型和语义数据模型是紧密相关的,有时被看做是数据库模型。在信息模型中,实

体 E 是一个对象或一组对象。实体把信息收集在其中,关系 R 是实体之间的联系或交互作用。有时在实体和关系之外,再加上属性。实体和关系形成一个网络,描述系统的信息状况,给出系统的信息模型。

信息建模和面向对象分析很接近,但仍有很大差距。在 E-R 图中,数据不封装,每个实体和它的属性的处理需求不是组合在同一实体中的,没有继承性和消息传递机制来支持模型。但 E-R 图是面向对象分析的基础。

4. 面向对象方法

面向对象的分析是把 E-R 图中的概念与面向对象程序设计语言中的主要概念结合在一起而形成的一种分析方法。在该方法中采用了实体、关系和属性等信息模型分析中的概念,同时采用了封装、类结构和继承性等面向对象程序设计语言中的概念。

3.4.5 需求分析过程

需求分析包括提炼、分析和仔细审查已收集到的需求,以确保所有的风险承担者都明白其含义并找出其中的错误、遗漏或其他不足的地方。需求分析的目的在于开发出高质量和具体的需求,这样就能做出实用的项目估算并可以进行设计、构造和测试。

需求分析的主要过程包括以下几个阶段。

1. 定义系统的边界

建立系统与其外部实体间的界限,明确接口处的信息流。

2. 分析需求可行性

分析每一个需求实现的可行性,确定与需求实现相关的开发风险。

3. 确定需求优先级

由于软件项目受到时间和资源的限制,一般情况下无法实现软件功能的每一个细节,因此需求优先级有助于开发组织和版本规划,以确保在规定的时间和预算内达到最好的效果。

4. 建立需求分析模型

建立需求分析模型是需求分析的核心工作,它通过建立需求的多种视图,揭示出需求的不正确、不一致、遗漏和冗余等更深的问题。

5. 创建数据字典

数据字典定义了系统中使用的所有的数据项及其结构,以确保客户和开发人员使用一致的定义和术语。

多年来,人们提出了许多分析建模的方法,其中占主导地位的是传统的结构化方法和目前流行的面向对象分析方法,将在后面的章节中分别介绍。

3.5 需求定义

需求定义是将对系统分析的结果用标准化的文档,即软件需求规格说明书(Software Requirement Specification, SRS)的形式清晰地描述出来,以此作为审查需求分析阶段工作完成情况的依据和设计阶段开展工作的基础。需求规格说明书是系统所有相关人员,包