

第 3 章 软件需求工程

学习目标

- 了解需求工程过程、需求评审与验证、需求变更控制；
- 理解软件需求规格说明；
- 掌握软件需求、需求获取技术。

软件需求是指用户对软件的功能与性能需求,就是用户希望软件能够做什么事情,完成哪些功能,达到哪些性能等。

在开发软件之前,必须先了解软件的需求,充分理解设计和使用这个软件的意图。软件需求工程的目的是定义软件所需要解决的问题。显然,一个软件开发项目不可能在初步识别要解决的问题之前启动,但是这些问题往往定义模糊。从这种情况来看,可以认为软件需求是要把一个定义不足和模糊的问题转换为一个定义良好而准确的问题,进而找到解决问题的方案。

3.1 软件需求工程概述

完成软件项目需求是软件生存周期中重要而关键的一步。其主要困难表现如下。

(1) 双方固有的矛盾。软件开发人员与用户的思维方式不同,因而用户与软件开发人员交流存在困难。所以在需求过程中,需求的供求双方经常会遇到双方不能达成共识或双方达成共识的内容有相当大的区别情况。

(2) 需求具有易变性和难以表述性。在软件项目中,40%~60%的问题都是在需求分析阶段出了问题。软件需求存在需求难以描述清楚,即难以表达。因为需求存在易变性和难以表达性,所以需求的获得需要有科学的分析方法和管理工作。

(3) 需求错误的高频性和修复的高成本性。在软件开发的各阶段,软件缺陷的总结如表 3-1 所示。

表 3-1 中,把软件中潜在缺陷的总数量定义为 1,这只是一种相对表示,与绝对数量无关。通过表 3-1 中的数据可以看出,剩余的缺陷中,需求缺陷最多,占总剩余缺陷的 30.87%(0.046/0.149),这些数据说明需求错误是软件项目开发中最常见的。

对于需求阶段就出现的错误,如果在软件项目进行到后期的时候才发现,那么修复费用昂贵,甚至会超出项目本身的费用。图 3-1 给出了软件项目生命周期的不同阶段修复

表 3-1 软件缺陷总结

缺陷来源	潜在缺陷	剩余缺陷	排除效率(%)
需求	0.2	0.046	77
设计	0.25	0.0375	85
编码	0.35	0.0175	95
建档	0.12	0.024	80
修复	0.08	0.024	70
合计	1	0.149	85.1

缺陷的相对成本,其中将编码阶段的修复成本定义为 1。

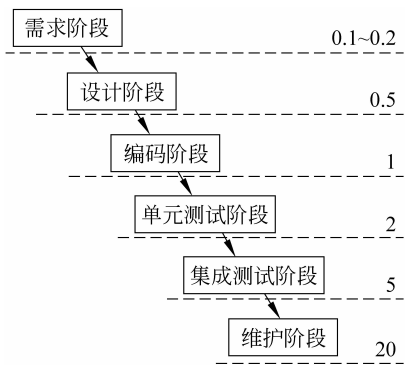


图 3-1 软件缺陷修复的成本

从图 3-1 中可看出,一个在需求阶段出现的错误,在维护阶段修复它的成本约是需求阶段修复成本的 100~200 倍。出现这种修复成本急剧上升的原因是如果需求错误在需求阶段就能发现的话,只需要重新进行规格说明,但如果直到维护阶段才发现的话,则需要重新进行规格说明、重新设计、重新编码、重新测试、重新建立文档等一系列工作。

因此,对于软件缺陷的发现和修复越早,则成本越低。然而需求阶段出现的错误往往很难发现,经常会延续到后面的阶段。这就造成了为克服需求错误而付出的高昂代价,同时也说明了做好需求

管理、减少需求错误的出现对于降低软件项目的成本至关重要。

3.1.1 软件需求分类

软件需求可分为下述几类。

1. 业务需求

业务需求说明软件系统必须完成的任务、实际业务或工作流程等。软件开发人员通常可从业务需求进一步细化出具体的功能需求和非功能需求。

2. 目标需求

目标需求反映组织机构或客户对系统和产品提出的高层次的目标要求,其限定了项目的范围和项目应达到的目标。

3. 功能需求

功能需求指开发人员必须实现的软件功能或软件系统应具有的外部行为。

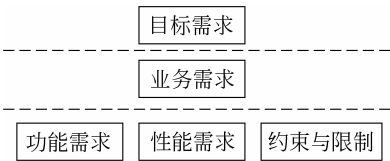
4. 性能需求

性能需求指实现的软件系统功能应达到的技术指标,例如响应时间、精度、用户数量和可扩展性等。

5. 约束与限制

约束与限制指软件开发人员在设计和实现软件系统时的限制,例如开发语言、数据库管理系统等。

由上述这些需求可以构成软件需求规格说明。
各种软件需求的层次关系如图 3-2 所示。



3.1.2 需求规格说明

图 3-2 各种软件需求的层次关系

需求规格说明是指软件所应满足的全部要求,并用文档方式完整和精确描述。全部要求是指软件系统必须提供的功能和性能、约束条件和限制。软件需求规格说明的一般格式如下。

- 1. 引言
 - (1) 编写目的(阐明编写需求说明书的目的,指明读者对象)。
 - (2) 项目背景(应包括: a. 项目的委托单位、开发单位和主管部门; b. 该软件系统与其他系统的关系)。
 - (3) 定义(列出文档中所用到的专门术语的定义和缩写词的原文)。
 - (4) 参考资料(可包括: a. 项目经核准的计划任务书、合同或上级机关的批文; b. 项目开发计划; c. 文档所引用的资料、标准和规范。列出这些资料的作者、标题、编号、发表日期、出版单位或资料来源)。
- 2. 任务概述
 - (1) 目标。
 - (2) 运行环境。
 - (3) 条件与限制。
- 3. 数据描述
 - (1) 静态数据。
 - (2) 动态数据(包括输入数据和输出数据)。
 - (3) 数据库描述(给出使用数据库的名称和类型)。
 - (4) 数据词典。
 - (5) 数据采集。
- 4. 功能要求
 - (1) 功能划分。
 - (2) 功能描述。
- 5. 性能需求
 - (1) 数据精确度。
 - (2) 时间特性(如响应时间、更新处理时间、数据转换与传输时间、运行时间等)。
 - (3) 适应性(在操作方式、运行环境、与其他软件的接口以及开发计划等发生变化时,应具有的适应能力)。
- 6. 运行需求
 - (1) 用户界面(如屏幕格式、报表格式、菜单格式和输入输出时间等)。

- (2) 硬件接口。
- (3) 软件接口。
- (4) 故障处理。
- 7. 其他要求(如可使用性、安全保密、可维护性和可移植性等)
- 8. 附录

制定软件的需求规格说明不仅是软件开发者的任务,而且用户也起着极其重要的作用。首先用户必须对软件功能和性能提出初步要求,并澄清一些模糊概念。然后软件分析人员认真了解用户的要求,细致地进行调查分析,把用户要求做什么的问题转换成一个完全的、精细的软件逻辑模型,并写出软件的需求规格说明,准确地表达用户的要求。

需求规格说明的特性如下:

- 完整性;
- 正确性;
- 可行性;
- 必要性;
- 无歧义性;
- 可验证性;
- 划分优先级。

3.1.3 需求工程概念

需求工程就是应用工程化的方法、技术和规格来开发和管理软件的需求。需求工程的目标是获取高质量的软件需求。需求工程突出了工程化原则,强调以系统化、条理化和重复化的方法进行软件需求的相关活动,从而增强了管理性和降低了需求开发的成本。需求工程由需求的开发活动和管理活动组成。需求工程的主要任务如下。

- (1) 确定待开发软件系统的用户,并获取用户的需求信息。
- (2) 分析用户的需求信息,并按需求类型分类,过滤掉非需求的信息。
- (3) 根据需求信息建立软件系统的逻辑模型和需求模型,确定非功能需求和约束条件及限制。
- (4) 根据收集的需求信息和逻辑模型编写需求规格说明及文档。
- (5) 评审需求规格说明。
- (6) 当需求变更时,对需求规格说明及需求变更进行管理。

3.1.4 需求工程过程

软件需求的开发和管理过程是由导出、确认和维护软件系统需求规格说明的一系列活动组成的,需求工程过程分为需求开发和需求管理两个阶段,如图 3-3 所示。

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。

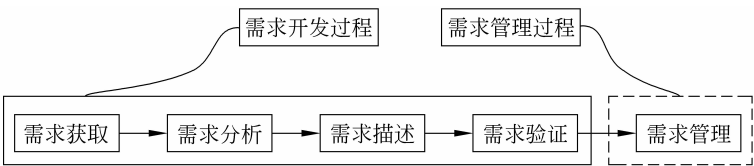


图 3-3 需求工程过程

2. 需求分析

对获得的用户需求信息进行分析、比较和综合，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和文档。

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和文档，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。

1~4 项是属于软件系统需求开发阶段的内容，需求开发阶段的主要工作是收集、归纳和整理用户的问题和要求，确定系统要做什么和应该做什么，然后将其明确，也就是分析用户意图，明确需求和约束条件的内容。在此之后，从逻辑上完整和严格地描述将要开发的软件系统，确保其能够反映用户的需求，这就是需求规范化的内容。

3.2 需求获取方法

需求工程的第一项活动是需求获取，这项活动的主要任务是获取用户的需求信息。需求获取是软件开发中最关键和最困难的活动，完成这项活动需要开发人员和用户密切合作，需要开发人员从用户提供的大量信息中理解和分析出用户真正的需求。需求获取过程的描述如图 3-4 所示。

1. 确定需求开发计划

本项工作的基本任务是确定需求开发的步骤，提出收集需求活动的具体安排和进度。应注意下述几点：

- (1) 为了能够充分利用时间和经费，只对与需求开发相关的工作安排进度和时间；
- (2) 在安排进度和时间时，充分考虑困难性，应按最艰难的情况安排；
- (3) 在安排进度和时间时，充分考虑灵活性，以便应对突

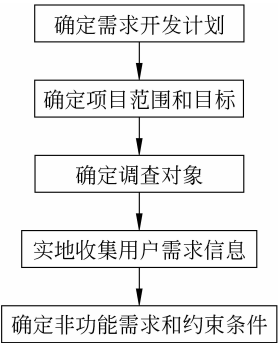


图 3-4 需求获取过程

变的情况；

(4) 在安排进度和时间时,应考虑到书写和整理需求规格说明及其文档所用的时间。

2. 确定项目范围和目标

项目目标主要指项目开发的目的和意义,以及软件系统的目标。项目的范围是指软件系统所涉及的各方面、应包括和不应包括的部分,与软件系统的目标需求密切相关。目标需求代表了需求层次中最高层次的需求,为软件系统定义了作用范围。

3. 确定调查对象

确定调查对象的基本任务是明确地确定来自不同层次的需求来源和用户,并将其分类。从图 3-1 中可以看出,软件的需求分为三个层次,因此可以按需求的层次来区分不同的用户。

- (1) 提出目标需求的用户；
- (2) 提出业务需求和功能需求的用户；
- (3) 软件开发人员。

4. 实地收集用户需求信息

实地收集需求信息阶段的任务是到现场实地调查和与用户交流,收集和理解用户需求信息。

(1) 实地调查的步骤：

① 向组织机构的负责人和高层管理人员调查,有利于对系统的宏观了解,这是目标需求的主要来源。

② 向部门负责人调查,这是功能需求和非功能需求的主要来源。

③ 向业务人员调查,因为业务人员十分了解自身工作的细节,所以此项调查可以获得具体功能和性能的需求信息。

(2) 实地收集需求信息的基本方法：

- ① 会议座谈；
- ② 书面咨询；
- ③ 应用用例。

(3) 需求信息分类：

① 目标需求：描述用户可获得的利润及与产品相关发展规划等信息。

② 用例说明：用户描述如何利用系统完成业务任务和如何实现用户要求的目标,应与用户共同讨论后确定。

③ 业务规则：是指一些活动只能在特定的条件下用一条业务规则表示。

④ 功能需求：描述了系统应该做什么。

⑤ 性能需求：是一种定性指标。

⑥ 外部接口需求：描述系统与外部的关系。

⑦ 限制：指合理限制设计者和程序员的选择条件。

⑧ 数据定义：描述一个数据项或一个复杂的业务数据结构的格式、允许值或默认值。

⑨ 解决方案：解决方法和步骤的描述。

5. 确定非功能性需求

非功能性需求是表明软件能否良好运行的定性指标。常用的非功能性需求如下：

- 可靠性；
- 可用性；
- 安全性；
- 互操作性；
- 易用性；
- 可维护性；
- 可移植性；
- 可用性；
- 健壮性。

3.3 需求分析的任务与原则

3.3.1 需求分析的任务

需求分析的基本任务是分析与综合已收集到的需求信息,通过分析找出需求信息内在联系和可能的矛盾,通过综合找出解决问题的方法并建立系统的逻辑模型。具体地说,需求分析的是提炼、分析和审查已收集到的需求信息,找出真正的和具体的需求,并确保所有相关人员都理解其含义。此外,通过建立软件系统的逻辑模型,发现和找出需求信息中存在的冲突、遗漏、错误或含糊的问题。

1. 绘制系统关联图

关联图是用于定义系统与系统外部实体间的界限和接口的简单模型。同时它也明确了通过接口的信息流和物质流。

2. 创建用户接口原型

当开发人员或用户不能确定软件需求时,开发一个用户接口原型(可能的局部实现),这样使得许多概念和可能发生的事更为直观明了。用户通过评价原型将使项目参与者能更好地相互理解所要解决的问题,并要找出需求文档与原型之间所有的冲突之处。

3. 分析需求可行性

在允许的成本、性能要求下,分析每项需求实施的可行性,明确与每项需求实现相联系的风险,包括与其他需求的冲突,对外界因素的依赖和技术障碍等。

4. 确定需求的优先级

应用分析方法来确定使用实例、产品特性或单项需求实现的优先级。以优先级为基础确定产品版本将包括的特性或需求。当允许需求变更时,在特定的版本中加入每一项变更,并在那个版本计划中做出需要的变更。

5. 为需求建立模型

需求的图形分析模型是软件需求规格说明的补充说明。它能提供不同的信息与关系,以有助于找到不正确的、不一致的、遗漏的和冗余的需求。这样的模型包括数据流图、实体关系图、状态变换图、对话框图、对象类及交互作用图。

6. 创建数据字典

数据字典是对系统用到的所有数据项和结构的定义,以确保开发人员使用统一的数据定义。在需求阶段,数据字典至少应定义客户数据项,以确保客户与开发小组是使用一致的定义和术语。分析和设计工具通常包括数据字典组件。

7. 质量功能调配

质量功能调配是一种高级系统技术,它将产品特性、属性与对客户的重要性联系起来。该技术提供了一种分析方法,以明确哪些是客户最为关注的特性。质量功能调配将需求分为期望需求和普通需求,期望需求客户或许并未提及,但如缺少,则感到不满意;普通需求实现了会给客户带去惊喜,但若未实现,也不会受到责备。

3.3.2 需求分析的原则

近年来已提出了许多软件分析与说明的方法。虽然各种分析方法都有其独特的描述,但总的看来,所有分析方法有共同适用的基本原则。

1. 必须能够表达和理解问题的数据域和功能域

所有软件定义与开发工作最终是为了解决数据处理问题,就是将一种形式的数据转换成另一种形式的数据。其转换过程必定经历输入、加工数据和产生结果数据等步骤。对于计算机程序处理的数据,其数据域应包括数据流、数据内容和数据结构。

数据流即数据通过一个系统时的变化方式。输入数据首先转换成中间数据,然后转换成输出结果数据。在此期间可以从已有的数据存储(如磁盘文件或内存缓冲区)中引入附加数据。对数据进行转换是程序中应有的功能或子功能,两个转换功能之间的数据传递就确定了功能间的接口。数据内容即数据项。数据结构即各种数据项的逻辑组织。

2. 按自顶向下、逐层分解问题

分解问题是把问题以某种方式分解为几个较易理解的部分,并确定各部分间的接口,从而实现整体功能。在需求分析阶段,软件的功能域和信息域都能做进一步的分解。这种分解可以是同一层次上的,称为横向分解;也可以是多层次的纵向分解。

把一个功能分解成几个子功能,并确定这些子功能与父功能的接口,就属于横向分解。但如果继续分解,把某些子功能又分解为小的子功能,某个小的子功能又分解为更小的子功能,这属于纵向分解。

3. 要给出系统的逻辑视图和物理视图

给出系统的逻辑表示(给出系统的逻辑视图)和物理表示(给出系统的物理视图)。这对系统满足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条件是必不可少的。进而确定做怎样的实现形式限制、功能限制及算法与数据限制。如数据字典,

就对数据流的条目、文件条目、数据项条目及加工条目进行了具体的规定和限制。

软件需求的逻辑视图给出软件要达到的功能和要处理数据之间的关系,而不是实现的细节。系统中检查库存的功能只关心库存文件的数据结构,而不关心在计算机中的具体存储方式。软件需求的逻辑描述是软件设计的基础。

软件需求的物理视图给出处理功能和数据结构的实际表示形式,这往往是由设备决定的。如一些软件靠终端键盘输入数据,另一些软件靠参数转换设备提供数据。分析员必须弄清系统元素对软件的限制,并考虑功能和信息结构的物理表示。

3.4 需求建模方法

需求建模的主要任务是根据待开发软件系统的需求,利用某种建模方法建立该系统的逻辑模型,以帮助软件开发人员检测软件需求的一致性、完整性、二义性和错误等。目前,已提出多种建模方法,它们具有下述共同的特性。

1. 提供描述手段

在软件开发过程中,开发人员间的有效交流是项目成功的关键之一,每个开发人员必须将工作结果以一定的形式记录下来。需求建模应该规定描述模型的手段,包括记录内容和用什么符号来表达等。

2. 提供基本步骤

需求建模需要规定基本实施步骤,确定每一步的目的、应注意的概念和需要掌握的必要信息,以及要产生的结果,辅助工具等。

建模方法主要包括结构化的需求建模方法和面向对象的需求建模方法。

3.4.1 结构化的需求建模方法

结构化的需求建模方法的基本特点是表达问题时尽可能使用图形符号的形式,设计数据流程图时只考虑系统必须完成的基本功能,不必考虑如何具体实现这些功能。

结构化的需求建模方法的基本思想是按照由抽象到具体、逐层分解的方法,确定软件系统内部的数据流、变换关系,并用数据流程图表示。

结构化的需求建模方法的描述手段由三部分组成:

- (1) 一套分层的数据流程图:主要说明系统由哪些部分组成,以及各部分之间的联系。
- (2) 一本词典:为数据流程图出现的每个元素提供详细的说明。
- (3) 其他补充材料:具体的补充和修改文档的说明。

结构化分析方法是面向数据流进行需求分析的方法。由 E. Yourdon 等人于 20 世纪 70 年代末提出和发展,至今已得到广泛应用。结构化分析方法的一些重要概念也渗透在其他开发方法中。

结构化分析方法使用数据流程图与数据字典来描述,面向数据流问题的需求分析适合于数据处理类型软件的需求描述。其核心思想是分解化简问题,将物理与逻辑表示分开,对系统进行数据与逻辑的抽象。具体来说,结构化分析方法就是用抽象模型的概念,按照

软件内部数据传递、变换的关系,自顶向下逐层分解,直到找到满足功能要求的所有可实现的软件为止。这种方法由于利用图形来表达需求,显得清晰、简明,易于学习和掌握。但其不足之处是不能表达复合逻辑的需求分析问题,不能详细表达加工。

软件需求的逻辑模型给出软件要达到的功能和要处理数据之间的关系,而不是实现的细节,即明确目标系统做什么问题。例如,一个商店的销售处理系统要从顾客那里获取订单,系统读取订单的功能并不关心订单数据的物理形式和用什么设备读入,也就是说无需关心输入的机制,只是读取顾客的订单而已。类似地,系统中检查库存的功能只关心库存文件的数据结构,而不关心在计算机中的具体存储方式。软件需求的逻辑模型是软件设计的基础。

一般来说,需求分析阶段包括下述几方面。

1. 确定对系统的综合需求

对系统的综合需求主要有系统功能需求、系统性能需求、运行需求、将来可能提出的需求。系统分析人员与用户协商,澄清模糊需求,删除无法做到的需求,改正错误需求。对于系统功能需求,应该划分出系统必须完成的所有功能。而系统性能需求包括响应时间、精确度指标需求、安全性等。运行需求集中表现为对系统运行时所处环境的需求,如软硬件运行环境限定需求等。最后,对于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需求。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做准备,以便于需要时能比较容易地进行这种扩充和修改,更有利于系统维护。

2. 分析系统的数据需求

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。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

就是在理解当前系统“怎样做”的基础上,抽取其“做什么”的本质。在物理模型中有许多物理因素,但随着分析工作的深入,有些非本质因素就显得不必要了,因而需要对物理模型进行分析,区分本质和非本质因素,去掉那些非本质因素就可获得反映系统本质的逻辑模型。综合上述两项分析的结果,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:首先确定目标系统与当前系统的逻辑差别;然后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;最后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

在经过需求分析阶段的工作后,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作出更准确的估计,在此基础上应该对开发计划进行修正。