

软件必然涉及应用,而且必须考虑与使用者有关的应用需求。软件需求分析即用来发现这种需求,由此挖掘用户价值。这是一项非常关键的、必须先于软件设计完成的工程任务,必须要收集用户需求意愿,面向用户建立软件需求分析模型,并依据需求模型确定软件产品技术规格,以此解答“软件能够用来做什么?”的用户提问。

本章要点:

- 需求分析任务。
- 获取用户需求。
- 建立需求模型。
- 软件需求验证。

5.1 需求分析任务

5.1.1 分析内容

软件需求指用户对软件的要求。用户通常会根据自身业务需要对软件提出要求,如要求财务软件系统能够按时自动生成财务分析报表,要求人力资源软件系统能够进行人力成本核算。

软件研发者需要根据用户需求研制软件。然而,软件研发者要能够研制出完全满足用户需求的软件并不是一件容易的事情。许多时候,研发者或许认为已经做到了用户所需,然而并不能得到用户认同。例如,研发者研制了一个功能看似全面、性能良好、交互界面也吸引人的软件,但用户却认为软件的业务处理流程与实际作业流程不一致。因此,面对一个研发者自认为很不错的软件,用户却是牢骚满腹,要求返工重做。

毫无疑问,软件需求问题影响了软件质量,降低了用户对软件的满意度,损害了软件开发机构的声誉。因此,在研制软件之前,必须先对软件做细致的需求分析,搞清楚用户有哪些方面的需求。

软件有来自用户诸多方面的需求,如功能需求、数据需求、性能需求、接口需求等。需求分析中需要清晰明辨并分类收集这些需求。

(1) 功能需求: 软件在服务上需要满足的要求,如定时发送通知、登录验证身份、年终汇总打印等。功能需求是保证软件可用的基本要求,需要优先考虑。

(2) 数据需求: 软件在数据上需要满足的要求,如数据表现格式、数据存储结构等。数据需求也是保证软件可用的基本要求,需要优先考虑。

(3) 性能需求：软件在时间与空间约束上需要满足的要求，如操作响应最大时间限制、数据传输最小速率限制、系统安装最小外存要求、系统运行最小内存要求等。

(4) 接口需求：软件在通信或交互上需要满足的要求，包括硬件接口需求、软件接口需求、应用接口需求。其中，基于交互界面的应用接口需求最被用户关注，直接影响用户对软件质量的评价。

5.1.2 分析过程

需求问题贯穿软件研发全过程。通常，系统前期分析时需要考虑有关软件的全局性需求框架。例如，来自前期分析的某系统全局性需求框架，①需要采用 B/S 结构以适应基于互联网的远程访问；②需要建立分级权限控制以满足用户的多层级分组应用；③需要建立数据中心以满足分布环境下的数据共享。

全局性需求框架将决定整个软件系统的体系结构。然而，要研制出符合用户应用需要的软件，不仅需要确定符合其需要的软件体系结构，还必须确定符合其应用需要的功能细节和数据细节。需求分析的目的就是全面获取这些需求细节。

前期分析中已确定的全局性需求框架是需求分析的工作基础。

以需求框架为线索进行更全面的用户需求分析，涉及获取用户需求、建立需求模型、定义软件规格等多项任务步骤，分析过程如图 5-1 所示。

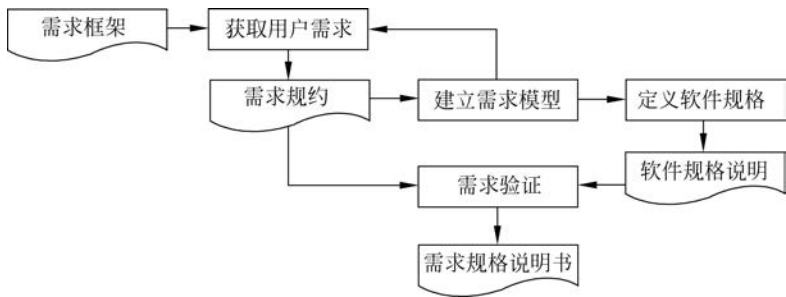


图 5-1 需求分析过程

5.1.3 任务承担者

需求分析需要软件开发者与用户共同参与，一般由开发者主导，但活动核心则是用户。开发者与用户都应该积极地扮演好自己的角色，因为只有这样，开发者与用户之间才能形成共识，产生出双方认同的需求约定。

需求分析需要对需求问题有较完整的求解，因此，不仅开发者与用户要达成需求约定，还需要考虑如何实现需求，这涉及软件规格，如功能规格、数据规格、性能规格，以使软件开发者在实现需求以及用户在确认需求时，能够有可信赖的技术与质量依据。

需求分析中开发者需要与用户交流、沟通、协商。然而，开发软件的技术专家习惯于从专业技术角度看待软件问题，更愿意使用技术性术语说明软件问题，可是用户并不能很好理解这些技术性术语。因此，开发者与用户之间可能出现沟通障碍。

需求分析需要既熟悉软件技术又熟悉用户业务的专业人员完成。系统分析师就是这样的需求分析专家，他们熟悉开发者与用户双方的术语，并有良好的需求分析技能，能够很好

地理解双方意图,协调双方意见。

实际上,系统分析师承担了将用户需求意愿转述为可用于约束软件设计实现的技术规格的职责,如图 5-2 所示。



图 5-2 系统分析师的职责

5.2 获取用户需求

软件需求源于用户,需求分析的首要任务就是获取用户需求。

系统工程中建立的全局性需求框架可为详细需求分析提供方向导航。实际上,一方面可基于高层需求框架逐项地获取用户的需求细节;另一方面诸多需求细节又在确认并充实需求框架。

用户需求细节应该越具体越明确越好。例如,需要打印工资与需要按职位分类打印工资,显然后一种需求描述更具体,更能反映需求细节。

分析者可直接从用户身上获取需求细节。需要注意的是,这些直接来自用户的需求往往是零散的,并可能存在歧义,或在技术上存在不合理性。因此,分析者还需要对收集到的需求进行筛选与分类。

此外,由于受软件认识的限制,用户需求意识大多非常模糊。一些表层需求,如报表格式、界面样式,或许能够直接形成意识;一些深度需求,如软件运行模式、数据存储方式、信息保密机制,则很难直接表达,而必须依靠分析人员的诱导才能挖掘出来。

5.2.1 识别用户

1. 用户特点

通常情况下,用户就是软件的使用人。然而,当用户作为一个与软件相关的抽象概念出现时,它就有着范围更广的外延,泛指系统以外一切可从软件获得服务的对象,包括软件使用机构、软件直接操作者、软件间间受益者,以及需要从软件获得服务支持的其他系统或设备,如图 5-3 所示。

软件需求源自用户。显然,如果用户是一个具体的人,如管理员、绘图员、打字员,开发者可以很直接地从这个可主动诉求的人的身上获取需求意愿。然而,如果用户是一个非具体的人,如机构、部门、其他系统、电子设备,开发者一般不能直接从它们身上获取需求,而是需要寻找用户代表(可代表用户的具有诉求能力的人),并从用户代表那里获取用户需求。

(1) 当用户是一个机构或一个部门时,该机构或部门的负责人可看成是用户代表。

(2) 当用户是一个有待集成的其他系统时,这个其他系统的创建者、使用者或销售者可看成是用户代表。

(3) 当用户是一个有待连接的外部设备时,这个设备的制造方代表或设备使用者可看成是用户代表。

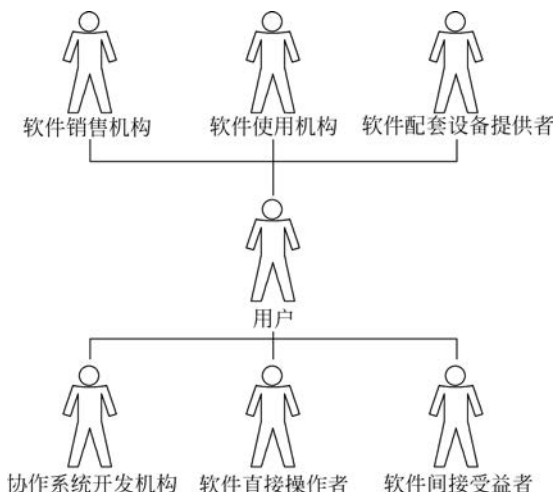


图 5-3 来自不同领域的用户

2. 用户分类

用户的业务环境及其需求大都是复杂的。现实情况往往是,开发者或许只是在开发一个规模并不算很大的软件系统,然而却不得不面对一个来自诸多领域的用户群,并需要面对来自于用户的各种各样的、复杂的需求愿望。显然,为方便分析用户需求,开发者有必要对用户进行分类。

对于业务管理系统,开发者通常可根据用户在业务组织中的位置将用户分为高层用户、中层用户与低层用户。

- (1) 高层用户是与系统相关的决策层用户。
- (2) 中层用户是与系统相关的部门层用户。
- (3) 低层用户是系统的最终操作者。

显然,不同层级的用户会有不同的软件需求。

高层用户所关注的可能是基于系统的业务发展,如新系统是否有利于提高工作效率,是否有利于拓宽业务面,是否有利于改善客户关系。

中层用户所关注的可能是基于系统的业务运作,如新系统是否能确保现有业务模式的正常运转,是否能更方便有效地提供所需要的业务数据。

低层用户所关注的则往往是实际操作,如新系统是否能提供更加人性化且更加方便、快捷的操作界面,是否能很快学会新系统的使用。

开发者也可根据用户与软件系统的亲密关系对用户进行分类,如核心用户、直接用户和间接用户。

- (1) 核心用户。软件系统管理员即为核心用户,有比较专业的软件使用能力,能够配置系统,能够对其他用户进行有效的授权控制。
- (2) 直接用户。软件系统一般操作员即为直接用户,能够比较熟练地操控系统。
- (3) 间接用户。与软件系统相关的部门负责人可看成是间接用户,他并不直接操控系统,但他需要从软件中获取如年度报表之类的数据。

5.2.2 从调查中收集用户需求

迄今为止,调查仍是收集用户需求的最基本途径。不同的软件系统将涉及不同的需求

问题,有不同的调查内容,并且需要采用不同的调查方法。

1. 调查提问

在需求调查中,分析人员需要依据有待开发软件的特点精心设计需求提问。

需求提问的一般顺序是:从软件系统的外部环境到软件系统的具体应用,然后逐步深入到软件系统的功能、性能、安全、人机交互等内部特征,从而比较全面地了解用户的需求意图。

需求提问还应考虑对用户的需求诱导,以启发用户的需求愿望。

例 5-1: 针对某制造厂生产管理系统的需求提问。

制造厂生产管理系统是一个与业务环境密切相关的系统,所涉及的业务因素有业务组织、业务范围、业务流程、业务数据,以下的需求提问即围绕这些因素展开。

- (1) 哪些活动将会影响产品生产?
- (2) 希望系统对哪些生产活动实施管理?
- (3) 谁将参与系统范围内的生产管理? 如部门、人员。
- (4) 是否可以画图说明生产管理流程?
- (5) 是否有使用软件系统管理产品生产的经验?
- (6) 原系统中的结果是否需要被新开发的生产管理系统保留下来? 如原系统的界面风格、原系统中的重要数据。
- (7) 生产管理系统是否需要导入其他系统中的数据? 如人力资源系统。
- (8) 生产管理系统产生的数据是否需要导出到其他系统? 如财务系统。

例 5-2: 针对某大厦智能监控系统的需求提问。

大厦智能监控系统与大厦需要提供的监控服务密切相关,所涉及的因素有大厦布局、服务设施、监控点、监控设备、监控目的等,以下问题即围绕这些内容展开。

- (1) 是否可以大致描述大厦布局?
- (2) 是否有比较全面的大厦布局图、结构图?
- (3) 大厦是否有监控中心?
- (4) 大厦将用到哪些监控设备?
- (5) 由谁提供这些监控设备?
- (6) 这些监控设备是否有配套驱动程序?
- (7) 这些监控设备是否有配套使用说明书?
- (8) 这些监控设备是否都需要连接到监控中心?
- (9) 希望如何进行防盗监控?
- (10) 希望如何进行防火监控?

2. 调查方法

不同的软件系统将需要用到不同的调查方法。一些常用的调查方法有访谈、座谈、问卷、跟班作业、收集资料等。

1) 访谈

访谈就是与个别用户面对面的交流沟通。

访谈可带来良好的沟通氛围,用户可比较自由地表达想法和愿望,访问者可从这种良好氛围中直接捕获到用户的需求意愿。

访谈还有利于消除用户因技术隔阂造成的需求疑惑,一些容易被用户误解的专门术语,

可通过访谈向用户做直接细致的需求解释。

然而访谈只能获取小范围调查,调查人数受限制,调查效率也比较低。因此,访谈形式一般只是用于调查对系统需要有全面把握的核心用户,或是用于调查将决定项目命运的高层用户。

2) 座谈

座谈就是开发者以会议形式邀集多方用户代表对需求问题进行商讨。

当所面临的需求问题需要考虑多部门业务协调时,座谈是一种很好的需求协商机制,可使各部门代表通过讨论来明确各自的业务边界。

有非正式与正式两种座谈形式。

非正式座谈一般用来建立人际交流,获取工作中所需要的情感沟通,使开发者与用户相互熟悉对方领域。非正式座谈中,参加人可比较自由地发表意见。

正式座谈则是一个比较规范的需求协商活动,有较严格的时间、议题限制,并需要产生比较正式的协商结论。正式座谈要求参加人严格按照议程顺序与议题发言,因此要有很好的准备,需要事先确定好会议地点、时间、参加人员、核心主题、相关问题,以及会议议程、目标等。

3) 问卷

问卷就是给用户分发需求调查问卷,一些主要的问卷形式是纸质问卷、网络问卷、电话问卷。

调查问卷有利于从数量众多的个体用户身上获取需求意愿。例如,开发者需要开发一个商业通用软件,为了使软件有一个较好的市场定位,就必须广泛地获取用户对软件的看法,这时可采取调查问卷方式进行需求调查。

4) 跟班作业

跟班作业为分析人员对用户业务活动的亲身体验。由于可获得最直接的需求体验,因此有利于对需求细节的把握。一些业务过程,如制造企业的生产过程,若仅凭来自用户的间接描述,则难免有细节遗漏,因此必须要有分析人员的跟班作业进行需求补充。

5) 收集资料

收集资料就是收集各种与待开发软件有关的文字和图片材料。例如,用户机构的组织结构图、业务流程图、业务规则、工作制度以及与业务有关的打印报表、数据资料等。

3. 整理调查结果

从调查中获得的用户需求往往是模糊和零散的。因此,分析人员还需要对调查到的用户需求进行整理,以使来自用户的需求结果能够清晰明了、便于理解。

如果用户需求中出现了需求歧义、需求冲突,或需求超越了现有技术条件,分析人员还须与用户讨论、协商、谈判,以使不科学、不合理的需求变为科学合理的需求。

实际上,经过整理的用户需求已不仅是原始用户需求,它还含有分析者的智慧,并体现为开发者与用户关于需求问题的共识。

5.2.3 建立需求规约

需求规约是开发者与用户就需求问题达成的约定。

开发者在投标软件项目时,就需要考虑与用户达成初步的需求约定。然而,这只是一个框架性规约,不涉及细则,只大致约定需要构建哪些子系统,需要具备什么功能,需要基于什么环境运行。

需求分析则需考虑规约细则。通过调查获取到的用户需求,应编入需求规约细则。

需求规约的特点如下。

(1) 面向用户: 实现与用户的沟通、交流,反映用户需求价值,使用易被用户理解的自然语言编制。

(2) 完整性: 应该全面反映用户需求,不仅需要说明有待开发的系统,还应该对有待开发系统的环境因素,如业务组织、设备、其他相关系统等给予必要说明。

(3) 逐步完善: 不可能一步到位,其中的诸多需求细节必然需要在以后的需求分析过程中逐步地验证、修正与补充。

需求规约是重要的需求分析成果,是需求建模与软件规格定义的基础。需求规约经需求验证后,将作为正式文档内容写进需求规格说明书,以反映用户需求价值。

例 5-3: “产品计划与生产管理系统”需求规约。

为减少产品积压,某制造厂决定基于产品市场订购实施产品生产,并决定委托某软件公司开发一个与该生产模式相适应的管理系统。

该生产管理系统要求具有以下功能:

- (1) 查询产品存量。
- (2) 签订产品合同。
- (3) 制订产品计划。
- (4) 制订材料计划。
- (5) 制订生产计划。
- (6) 监督生产进度。
- (7) 验收产品计划。

系统将应用于该制造厂的市场部、生产部、材料部。

(1) 市场部。

- 负责签订产品合同。
- 依据产品合同、产品存量状况,制订产品计划。
- 验收产品计划。

(2) 生产部。

- 依据产品计划,制订材料计划、生产计划。
- 依据生产计划,监督生产进度。

(3) 材料部。

- 依据材料计划,进行原材料配置。

例 5-4: “大厦智能监控系统”需求规约。

某大厦需进行智能防火、防盗监控,大厦已在需要监控的位置安装了红外线感应器,并且这些红外线感应器都已与监控中心计算机连线。现需要开发一个运行于监控中心计算机的安全监控系统,以实现当红外线感应器捕获到安全事件时,系统能够自动报警的功能。

该安全监控系统要求具有以下方面的功能。

(1) 系统配置功能。

系统管理员有权配置系统。

- 可开启或关闭整个系统。

- 可开启或关闭某些监控点。
- 可开启或关闭某些监控事件。
- 有 1、2、3 共 3 个报警级别,每级报警电话号码可分别设置。
- 可对监控点或监控事件设置报警级别。

(2) 系统报警功能。

- 当红外线感应器监控到安全事件时,安全监控系统会记录这个事件,事件现场会响铃报警,并且系统会逐级拨打报警电话。
- 报警电话会通知安全事件发生的地点和事因。
- 事件现场的响铃报警必须现场关闭才会终止。
- 设置的报警电话在响铃报警之后,将逐级地(每隔 20 秒上升一级)拨打报警电话,直到有其中的某一级报警电话被接听为止。如某监控点设置了三级报警,则在响铃报警之后,首先拨打一级报警电话;若一级报警电话 20 秒无人接听,将拨打二级报警电话;若二级报警电话 20 秒无人接听,将继续拨打三级报警电话;若这个过程中有一个报警电话被接听,则终止拨打报警电话。

5.3 建立需求模型

建立需求模型,也就是使用图形方式描述需求问题。通常,图形方式直观,有利于与用户进行需求探讨,有利于对用户需求进行抽象,有利于从用户需求到软件规格的过渡。

从获取用户需求到建立需求模型需要交替进行。这也就是说,抽象模型的创建往往不能一次完成,而是需要多次反复,因为只有这样,才能使用户需求获得最接近于现实的模型抽象。

需求分析通常需要从业务、功能、实体、行为等几个方面建立模型。

(1) 业务模型。业务模型是对用户业务的抽象,用来说明系统所面临的业务环境,如业务边界、业务流程、业务关系。常用建模工具有业务流程图、用例图、活动图。

(2) 功能模型。功能模型是对软件功能的抽象。建模内容有功能组成、功能过程。常用建模工具有功能树图、数据流图。

(3) 实体模型。实体模型是对与系统有关的现实实体及其关系的模型抽象。常用建模工具有实体关系图、类关系图。

(4) 行为模型。行为模型是对系统与环境之间交互的模型抽象。常用建模工具有活动图、状态图、时序图。

需求分析中首先需要考虑的是用户的业务需求。

一个系统尽管功能全面、性能优良,但如果与用户业务冲突,就没有使用价值,而要搞清楚用户业务,就需要建立业务模型。

5.3.1 业务域模型

业务域模型用于划分业务及其归属。来自 UML 的用例图能够用来划分业务域,并说明业务归属。每个椭圆用例图符用来表示一项业务,人形参与者图符用于表示用户,连线则用于表示业务归属于某个用户。图 5-4 所示是某“产品计划与生产管理系统”业务用例图,用于表达系统有哪些业务,谁将关联这些业务。

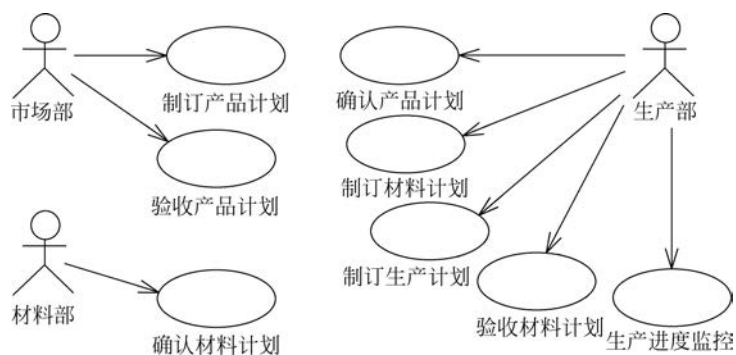


图 5-4 某“产品计划与生产管理系统”业务用例图

显然,业务域模型中的业务应有清晰的边界。因此,如果某两项业务中有共同的并可分离的业务单元,则两项业务的共同部分有必要作为专项业务抽取出来,并进行专门描述。

5.3.2 业务流模型

业务流模型用于表达业务活动步骤。传统流程图可说明业务流程,来自 UML 的活动图也可用来说明业务流程。图 5-5 所示是某“产品计划与生产管理系统”业务活动图。

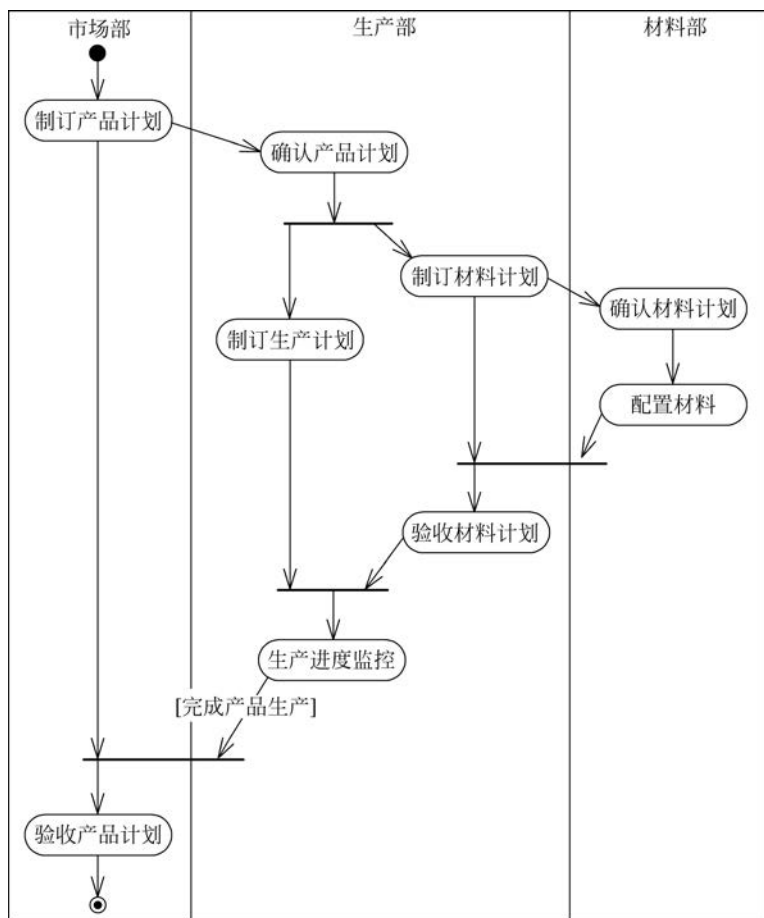


图 5-5 某“产品计划与生产管理系统”业务活动图

活动图有较强的建模能力,不仅说明活动顺序,还说明由一项活动到另一项活动的逻辑跳转。

业务层面的活动模型一般不涉及业务的局部逻辑控制,只反映业务的基本流程,其中的活动通常代表一个业务子项。

业务活动图还需考虑的是执行者,以反映业务归属。通常情况下,一个将经历多个业务子项的全局业务过程会涉及多个执行者(承担任务的部门或个人)之间的任务协助,因此也就需要考虑各业务子项的具体执行者是谁。

这些活动执行者通常可与用例图中的参与者对应,在活动图中一般使用泳道表示,如图 5-5 中的市场部、生产部和材料部。

5.4 定义与验证软件规格

5.4.1 软件规格定义

在完成软件需求建模后,可根据需求模型定义软件规格。

为使软件创建时有比较完整的规格依据,软件中的各项需求,如功能、数据、性能、接口等,都需要进行规格定义。

(1) 功能定义。功能定义是系统中功能成分的规格说明,主要规格要素有功能特征、功能边界、数据输入输出、异常处理等。

(2) 数据定义。数据定义是系统中数据成分的规格说明,主要规格要素有数据名称、别名、组成元素、出现位置、出现频率等。

(3) 性能定义。性能定义是系统中的性能指标的规格说明,如响应时间、传输速率、存储容量等。

(4) 接口定义。接口定义是系统中接口的规格说明,包括硬件接口、软件接口和应用接口。

5.4.2 软件需求验证

需求验证是对需求分析成果(如需求规约、软件规格定义)的检查与确认。通常可从以下几方面进行需求验证。

(1) 有效性验证:检查并确认需求规约或规格定义中的每项条款都是为满足用户应用而建立的,其中没有无用的多余条款。

(2) 一致性验证:检查并确认需求规约或规格定义中的每项条款都是相容的,相互之间没有内容冲突。

(3) 完整性验证:检查并确认需求规约或规格定义所给出的需求集合已对用户目前应用有了较完整的需求表述。

(4) 现实性验证:检查并确认需求规约或规格定义中的每项条款都是可最终实现的软件需求。

(5) 可检验性验证:检查并确认需求规约或规格定义中的每项条款都可被用户体验或检测。

5.4.3 通过原型验证用户需求

用户需求首先体现在需求规约上,它是直接面向用户的。通常情况下,为了使需求规约能更加准确地表达用户的需求意愿,其中的各项需求说明有必要让用户做进一步的验证确认。然而,有很多因素在干扰用户的需求验证确认,如用户需求的不稳定、用户对软件认识的局限。现实情况往往是,分析者已向用户多次解释需求规约,然而用户并不能很好地理解规约。需要引起重视的是,用户需求验证的艰难必将影响软件开发进程。实际上,需求分析后续工作不得不因验证的艰难而停顿下来,导致软件的设计、编码工作无法启动。

许多成功的软件项目经验告诉我们,需求原型有利于改善需求分析工作环境。需求原型可给用户带来直观的需求感受。一些可被用户直接看到的需求可通过原型验证,如界面、报表、数据查询的需求验证。

需求原型大多是抛弃型原型,无须考虑正常使用,因此可考虑通过软件快速生成工具创建需求原型。基于原型的需求验证过程如图 5-6 所示。

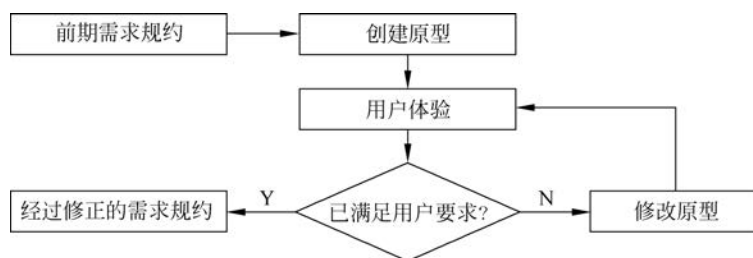


图 5-6 基于原型的需求验证过程

需要指出的是,发挥需求原型的价值还依赖于分析者的有效把握。通常情况下,分析者应该主动、有针对性地向用户演示原型,并在演示原型的同时逐项说明需求规约。

分析者还有必要指导用户与原型进行交互,以使用户能从原型中获得更真切的需求体验,并认真记录用户由原型体验而诱发出来的新的需求认识。

5.4.4 通过评审验证产品规格

需求评审则是一种传统的正式的需求验证机制。

通常需要有一个专门的评审小组进行需求评审。这应该是一个由各方面专家或代表(如软件分析师、软件工程师、领域专家、用户代表)组成的评审小组,他们将一同检查需求规约和软件规格中的各项需求说明。

可从以下 8 方面对需求进行评审。

(1) 一致性: 检查是否有需求冲突,如功能之间是否有相互矛盾的规约说明。

(2) 有效性: 检查每一项需求是否都符合用户的实际需要。

(3) 完整性: 检查是否有需求遗漏,需求规约是否已很完整地反映了用户的需求意愿。

(4) 现实性: 检查各项需求是否都能通过现有技术给予实现、用户是否可在开发出来的软件中看到需求结果。

(5) 可检验性: 检查各项需求是否都有适合于用户的检测方法;当软件交付用户使用时,用户是否可自行进行需求检验。

- (6) 可读性: 检查需求规约是否具有可读性,是否能够被用户轻松理解。
- (7) 可跟踪性: 检查是否清楚记录了各项需求的出处。
- (8) 可调节性: 检查是否能够应对可能出现的需求变更。

5.5 需求规格说明书

需求分析以建立需求规格说明书为任务目标,它是需求分析的成果体现,是需求分析阶段需要交付的文档。

几乎所有与软件有关的人员,如用户、项目管理人员、系统开发人员、系统测试人员,都需要阅读这份文档。

- (1) 用户: 按照需求规格说明书对软件系统进行验收。
- (2) 项目管理人员: 根据需求规格说明书制订项目详细开发计划,安排项目进程。
- (3) 系统开发人员: 以需求规格说明书为依据进行系统设计与实现。
- (4) 系统测试人员: 以需求规格说明书为依据进行系统有效性测试。

下面是需求规格说明书的参考格式。

1. 引言

- 1.1 编写目的(阐明编写本需求规格说明书的目的,指出读者对象)
- 1.2 项目背景(列出本项目委托单位、开发单位和主管部门,说明该软件系统与其他系统的关系)
- 1.3 定义(列出本文档中所用到的专门术语的定义和缩写词的原意)
- 1.4 参考资料(说明报告中引用的资料,包括本项目经核准的计划任务书、合同或上级机关的批文,项目开发计划,本文档需要引用的论文、著作,需要采用的标准、规范)

2. 系统概述

- 2.1 系统定义(说明系统目标,定义系统边界,描述系统业务规则。可使用顶层数据流图或用例图进行描述)
- 2.2 处理流程(说明系统数据或业务处理流程,可使用数据流图或活动图描述)
- 2.3 运行环境(如硬件设备、操作系统、支撑软件、数据环境、网络环境)
- 2.4 条件与限制(如软件平台限制、硬件设备限制、开发规范或标准的限制)

3. 功能需求

- 3.1 功能划分(可使用功能层次图说明系统功能组织结构)
- 3.2 功能描述(对功能层次图的每一个功能点进行细节说明)

4. 性能需求

- 4.1 数据精确度(输出数据的精确位数)
- 4.2 时间特性(如操作响应限时、数据更新限时、数据转换与传输限时等)
- 4.3 适应性(当操作方式、运行环境等发生变化时,系统应具有适应能力)

5. 运行需求

- 5.1 用户界面(如屏幕格式、报表格式、菜单格式等)
- 5.2 硬件接口(如串行接口、并行接口、USB 接口,以及其他能支持的设备接口)
- 5.3 软件接口(与其他需要协调工作的软件的接口)
- 5.4 通信接口(如网络连接器件、网络通信协议)
- 5.5 故障处理(说明当出现软件、硬件故障时的应急处理)

6. 其他需求(可使用性、安全性、可维护性、可移植性等)

7. 数据描述

- 7.1 静态数据(建立实体关系模型说明静态数据结构)
- 7.2 动态数据(输入、处理、输出中的数据)

小 结

1. 分析任务与过程

需求分析是为有效解决用户需求问题而需要进行的一项工程活动,所需考虑的需求问题有功能需求、数据需求、性能需求和接口需求。

开发者与用户都将参与需求分析活动。开发者承担分析任务,但活动核心则是用户。

需求分析任务需要由熟悉用户业务的系统分析师承担。

需求分析的步骤是获取用户需求、建立需求模型、定义软件规格、进行需求验证。

2. 获取用户需求

用户泛指一切可从软件获得服务的对象。可以是某个人,也可以是人以外的机构、部门、其他系统或设备。

可通过调查而获取用户需求。然而,有效的需求收集则依赖于有效的调查提问,并依赖于合适的调查方法。一些常用调查方法是访谈、座谈、问卷、跟班作业、收集资料。

来自用户的需求将体现为需求规约,用于表达用户的软件价值。

3. 建立需求模型

需求模型是用户需求问题图解,一些常用模型有业务树、用例图、活动图。其中业务树是结构化需求建模,用例图是系统业务举例,活动图则反映系统工作流程。

4. 软件需求验证

需求验证是指对需求分析成果的检查与确认。主要的需求验证内容有有效性验证、一致性验证、完整性验证、现实性验证和可检验性验证。

可通过需求原型确认或需求评审实现需求验证。

习 题

1. 什么是软件需求? 有哪些方面的软件需求?
2. 软件往往因不能满足应用需求而遭遇用户抱怨。对此,如果你是软件开发者,你有什么看法? 并有何解决措施?
3. 通常认为,系统分析师是需求分析专家。作为专家,应该具备哪些素质?
4. 请对需求分析基本过程做出说明。
5. 什么是软件用户? 有哪些方面的软件用户?
6. 调查仍是收集用户需求的最主要途径。常用的调查手段有哪些?
7. 需求分析中可建立哪些方面的需求模型? 分别有什么用途?
8. 业务活动建模中,泳道代表什么?
9. 需求分析中涉及哪些方面的需求验证?
10. 如何通过原型进行需求验证? 如何通过评审进行需求验证?