

需求类文档写作

3.1 需求概述

作为技术人员,大家更多关注的是技术,但软件需求在很大程度上决定了软件是否正确,需求确定后不管如何实现,功能和质量给客户直接带来的价值远远比技术直接带来的价值要高。因此,做正确的事比正确地做事更重要。错误需求带来的问题一直是各个软件公司项目失败的首要原因,因为获得需求是个复杂的过程,要在实践中不断地学习,提高需求分析的能力。需求有以下三个层次。

1. 业务需求

描述客户的高层次目标,通常问题定义本身就是业务需求的表征。这种目标通常体现在两个方面。

(1) 问题:解决企业/组织运作过程中遇到的问题,如设备管理混乱、用户投诉量大、客户流失率高等。

(2) 机遇:抓住外部环境变化所带来的机会,以便为企业带来新的发展,例如电子商务、网上银行、物联网等。

业务需求就是系统目标,它是以业务为导向、指导软件开发的高层次需求。这类需求通常来自高层,例如项目投资、购买产品的客户、实际用户、市场营销部门或产品策划部门。业务需求从总体上描述了为什么要开发系统(why),组织希望达到什么目标,一般在可行性研究报告中反映,也可使用前景和范围(vision and scope)文档来记录业务需求,这份文档有时也被称做项目章程(Project Charter)或市场需求(Market Requirement)文档。组织愿景是一个组织对将使用的软件系统所要达成的目标的预期期望,如“希望实施 CRM 后公司的客户满意度达到 90%以上”就是一条组织愿景。

2. 用户需求

用户需求是指用户要使用产品完成什么任务,通常是在问题定义

的基础上进行用户访谈、调查,对用户使用的场景进行整理,从而获得来自用户角度的需求。用户需求必须能够体现软件系统将给用户带来的业务价值,或用户要求系统必须完成的任务,也就是说用户需求描述了用户能使用系统来做些什么(what),这个层次的需求是非常重要的。

作为需求捕获阶段的主要产物,用户需求主要具有以下特点:

(1) 零散。用户会提出不同角度、不同层面、不同粒度的需求,而且常常是以一句话形式提出的,如通过电话、短信等非正式方式提出的需求。

(2) 相互矛盾。由于不同用户处于企业/组织的不同层面,可能会出现盲人摸象的情况,导致需求的片面性。

因此,还需要对原始需求进行分析和整理,从而得出更加精确的需求说明。用例是表达用户需求的一种有效途径。

3. 软件需求

由于用户需求具有零散、片面的特点,因此需求分析人员还需要对其进行分析、提炼、整理,从而生成可指导开发的、更准确的软件需求,软件需求是需求分析与建模的产物。

软件需求是需求的主体,它是设计具体解决方案的依据(how),其数量往往比用户需求高一个数量级。这些需求记录在软件需求规格说明(Software Requirements Specification, SRS)中。SRS完整地描述了软件系统的预期特性,SRS一般被当作文档保存,设计、实现、测试、质量保证、项目管理以及其他相关的项目过程都要用到SRS。

3.2 软件需求的分类

软件需求可分为功能需求、质量需求、约束条件三种类型,质量需求和约束条件也叫非功能需求。

1. 功能需求

功能需求规定必须在产品中实现的软件功能,用户利用这些功能来完成任务,满足业务需求。

对于功能需求而言,最关键的是如何对其进行组织,否则一句话的描述就会十分分散,很难保证开发人员逐一理解和满足这些要求。

在传统的方法论中,会以系统→子系统→模块→子模块的层次结构来组织,和程序的结构相对应,但这样会割裂用户的使用场景。为了解决这个问题,现代需求理论更加强调需求分析人员从用户的角度将系统理解成一个黑盒子,从横向的使用视角来整理需求。

2. 质量需求

质量需求不同于产品的功能描述,它从不同方面描述产品的各种特性。这些特性包括可用性、可移植性、性能、安全等,它们对用户或开发人员都很重要。

质量需求描述有两个常见问题。

(1) 信息传递的无效性：在很多需求规格说明书中，会通过一个名为性能需求的小节来说明非功能需求，列出诸如高可靠性、高可用性、高扩展性等要求。但是很多开发人员根本就不看这些内容，因为这样的定性描述缺乏判断标准，故这种信息传递方法是无效的。

(2) 忽略了质量需求的局部性：经常会看到诸如“所有的查询响应时间都应该小于10s”的描述，但是当用户查询的是年度统计数据时，这样的需求是较难实现的，因此开发人员就会忽略和不理睬这样的需求，最终的结果就是导致它成为了摆设。因此更科学的做法是利用具体的应用场景来描述。

3. 约束条件

约束条件限制了架构师设计和构建系统时的选择范围，这看起来很简单，但是如果不了解它的类型，很可能会导致在收集此类信息时出现遗漏的现象，约束条件一般有五种。

(1) 非技术因素决定的技术选型：对于软件开发而言，有些技术选型并非由技术团队决定，而会受到企业/组织实际情况的影响，例如必须采用某种数据库系统等。

(2) 预期的运行环境：架构师在决定架构、选择实现技术时会受到实际的软硬件环境的影响，如果忽略了这方面的因素会给项目带来一些不必要的麻烦。

(3) 预期的使用环境：除了系统的运行环境，用户的使用环境（使用场合、软硬件环境等）也会对软件的开发产生很大的影响。

(4) 社会限制：如智能交通系统要适应车辆可能出现单双号的限制、视频播放网站要支持内容分级的限制等。

(5) 法律限制：如必须使用正版软件等。如果不愿付出相应成本，则可以选择开源的软件。

约束条件的表现形式多样，例如，用户界面要求、联机帮助系统要求、法律许可、外购软件，以及操作系统和开发工具等，因此在需求阶段就应注意搜集此类信息。如果是在设计阶段有补充，可以修改需求规格说明书，也可将其写到软件需求规格说明书的补充规约中。

3.3 需求过程

3.3.1 需求分析

1. 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。需求分析所要做的工作是深入描述软件的功能和质量要求，弄清软件设计的限制条件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。

通常软件开发项目是要实现目标系统的物理模型，即确定待开发软件的系统元素，并将功能和数据结构分配到这些系统元素中，它是软件实现的基础。

需求分析的任务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。需求分析阶段的任务包括下述几方面。

(1) 确定对系统的综合需求。

(2) 分析系统的数据需求。

系统的数据需求是由系统的信息流归纳抽象出数据元素的组成、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。

就是在理解当前系统的基础上,抽取其“做什么”的本质。

(4) 修正系统开发计划。

(5) 建立原型系统。

2. 需求分析的步骤

(1) 调查研究。

(2) 分析与综合。

应注意下述两条原则:第一,在分层细化时必须保持信息连续性,也就是说细化前后对应功能的输入输出数据必须相同;第二,当进一步细化将涉及如何具体地实现一个功能时,也就是当把一个功能进一步分解成子功能,并将考虑为了完成这些子功能而写出其程序代码时,就不应该再分解了。

(3) 书写文档。

(4) 需求分析评审。

3. 需求分析的原则

(1) 必须能够表达和理解问题的数据域和功能域。

(2) 按自顶向下方法逐层分解问题。

4. 需求分析方法

大多数的需求分析方法是由数据驱动的,数据域具有三种属性:数据流、数据内容和数据结构。通常,一种需求分析方法总要利用一种或几种属性。

需求分析方法具有以下共性。

(1) 支持数据域分析的机制;

(2) 功能表示的方法;

(3) 接口的定义;

(4) 问题分解的机制以及对抽象的支持;

(5) 系统抽象模型。

5. 需求规格说明与评审

需求规格说明从功能、质量和约束条件三个维度说明系统的要求。需求分析的评审必

须从一致性、完整性、现实性和有效性 4 个不同角度验证软件需求的正确性。

3.3.2 需求过程的管理

需求管理的目的是为软件项目的需求在客户和项目组之间建立并维护一个协议。

1. 对需求过程进行管理的要求

- (1) 应为需求管理提供足够的资源和资金。
- (2) 对履行需求管理活动的人员提供相应的培训。
- (3) 需求应被文档化。
- (4) 需求评审时项目组和其他相关组的成员都要参与评审。
- (5) 在需求管理过程中,为了在软件开发生命周期中的不同阶段进行需求跟踪,应当建立和维护需求跟踪矩阵。
- (6) 项目组要将需求作为软件计划、工作产品和活动的基础。
- (7) 为了决定需求管理的活动状态,要执行量度,并对需求管理的状态及时汇报和评审。
- (8) 需求管理的活动应该由项目经理定期地组织评审或以事件驱动进行评审。
- (9) SQA 代表定期地或事件驱动地评审需求管理过程的执行情况和工作产品的质量。

2. 获取需求的过程

1) 用户需求调查及定义

- ① 需求调查。
- ② 分析需求信息。
- ③ 撰写用户需求调研报告。
- ④ 细化并分析用户需求。
- ⑤ 撰写产品需求规格说明书。

2) 需求确认

- ① 非正式需求评审。
- ② 正式需求评审。
- ③ 获取需求承诺。

3) 需求跟踪

- ① 建立和维护需求跟踪矩阵。
- ② 查找需求与后续工作成果的不一致性。
- ③ 消除需求与后续工作成果的不一致。

4) 需求变更控制

- ① 需求变更申请。
- ② 审批需求变更申请。
- ③ 更改需求文档。
- ④ 重新进行需求确认。

3.3.3 需求获取的流程

1. 系统需求调查

需求分析人员调查系统需求,随时记录调查过程中所获取的需求信息,具体方式可以是:

- (1) 访谈;
- (2) 头脑风暴;
- (3) 实地考察;
- (4) 问卷调查;
- (5) 原型制作。

需求分析人员分析已经获取的需求信息,消除错误,归纳总结共性的用户需求并撰写《系统需求调研报告》,它是反映系统需求及其起源的文档,同时也是进行下一步需求调研(以澄清现有需求中的不完整性和不一致等问题)和需求分析的基础。需求的获得是一个不断积累和完善的过程,当需求存在不完整性、二义性、不一致性、不可实现和不可测时,需要继续进行需求调研活动以纠正上述问题。

2. 需求分析和定义

根据所形成的系统需求调研报告,需求分析人员与系统架构设计师配合,综合分析需求的优先级、可实现性、风险和项目研发的前景来确定项目的范围,并产生需求文档。需求文档包括:

- (1) 需求规格说明书。

根据软件需求规格说明书模板撰写软件需求规格说明书,它阐述系统必须提供的功能、质量以及所要考虑的限制条件,是系统的范围定义,需提供给用户确认。

- (2) 用例图和用例使用场景。

在已获取的功能需求基础上使用 UML 建立用例图,根据用例使用场景模板对用例进行扩展,撰写用例使用场景。

3. 需求确认

项目经理先在内部组织项目开发小组成员进行非正式的需求评审,以消除明显的错误和分歧。然后项目经理邀请客户和最终用户一起评审需求文档,尽最大努力使需求文档正确无误地反映用户的真实意愿。

当需求文档通过正式评审后,开发方负责人(项目经理)和客户对需求文档做书面承诺(将需求文档作为合同的附件或者直接在需求评审报告上签字)。

4. 需求跟踪

项目经理负责指定项目开发小组相关成员根据需求跟踪矩阵模板建立需求跟踪矩阵,以反映需求与后续工作成果之间的关系。当本工作成果或前续工作成果发生变更时,项目开发小组相关人员及时更新需求跟踪矩阵。需求管理流程如图 3-1 所示。

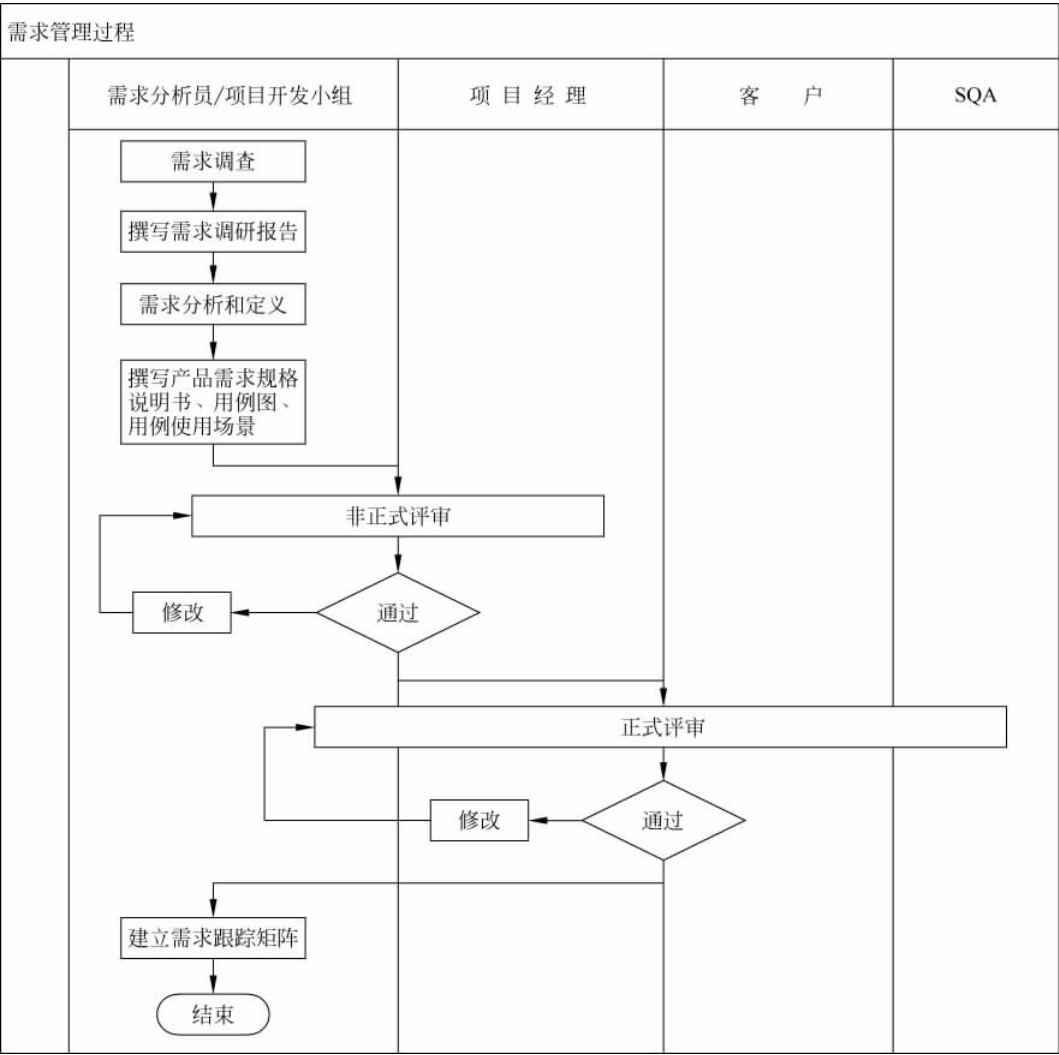


图 3-1 需求管理流程

需求过程的输入：
 客户提供的需求或工作说明。

输出：
 软件需求规格说明书；
 需求评审报告；
 需求跟踪矩阵。

3.3.4 需求管理的角色

控制机制：
 (1) 项目经理定期或以事件驱动的方式来评审需求。

- (2) 需求在基线前要进行评审。
- (3) SQA 参与需求文档评审,评审需求活动是否遵循需求管理过程。
- 需求管理角色表如表 3-1 所示。

表 3-1 需求管理角色表

角 色	职 责
项目经理	组织需求文档的非正式评审和正式评审,并做书面承诺 为项目指定适当的需求分析员 定期检查需求文档与后续工作成果的不一致性,通报相关人员加以处理,跟踪处理结果 提出需求变更申请
需求分析员	需求调研 需求分析 建立需求文档 建立需求跟踪矩阵
项目开发小组	参与需求文档的非正式评审和正式评审 解决由于需求变更而带来的问题
客户(公司客户代表)	参与需求文档的非正式评审和正式评审,并做书面承诺 提出需求变更申请
SQA	评审需求管理及开发过程中的各项活动

3.4 需求说明书的撰写要求

3.4.1 需求文档的文字叙述要求

软件需求说明书的编制是为了使用户和软件开发者双方对该软件的初始规定有一个共同的理解,使之成为整个开发工作的基础,因此需求的描述要清晰,让用户也容易理解,在软件需求文档中进行叙述时有以下几点建议:

- 保持语句和段落的简短。
- 采用主动语态的表达方式。
- 编写具有正确的语法、拼写和标点的完整句子。
- 使用的术语与词汇表中所定义的应该一致。
- 需求陈述应该具有一致的样式,例如“系统必须”或者“用户必须”,并紧跟一个行为动作和可观察的结果。例如,“药品管理子系统必须显示一张所请求的仓库中有存货的药品清单。”
- 为了减少不确定性,必须避免模糊的、主观的术语,例如用户友好、容易、简单、迅速、有效、支持、许多、最新技术、优越的、可接受的和健壮的。当用户说“用户友好”或者“快”或者“健壮”时,应该明确它们的真正含义并且在需求中阐明用户的意图。当用户说明系统应该“处理”、“支持”或“管理”某些事情时,应该能理解和表达用户的想

法。含糊的语句表达将引起需求的不可验证。

- 避免使用比较性的词汇,例如提高、最大化、最小化和最优化,应定量地说明所需要提高的程度或者说清一些参数可接受的最大值和最小值。

必须唯一地标识每个需求。对必须提交给用户的软件功能,应列出与之相关的详细功能需求,使用户可以使用所提供的功能执行服务或者使用所指定的使用实例执行任务。应描述产品如何响应可预知的出错条件和非法输入或动作。

3.4.2 对用例说明的要求

随着 UML 的日益普及,用例(Use Case)分析技术也在需求的功能说明中广泛被采用。但是也有许多团队在使用该技术时,只画出了用例图,而缺少用例说明,这是一个严重的误区。

用例的事件流描述应该说明系统内发生的事情,说明角色的行为及系统的响应,而不是事件发生的方式与原因。如果进行了信息交换,则需指出传递的具体信息。例如,只表述角色输入了客户信息就不够明确,最好明确地说角色输入了客户姓名和地址。当然也可以通过项目词汇表来定义这些信息,使得用例中的内容被简化,从而不至于让用例描述陷入过多的细节内容。

如果存在一些相对比较简单备选流,只需少数几句话就可以说清楚,那么也可以直接在这一部分中描述。但是如果比较复杂,还是应该单独放在备选流小节中描述。

一幅图胜过千言万语,因此建议在事件流说明中,除了叙述性文字之外,还可以引用 UML 中的活动图、顺序图、协作图、状态图等手段,对其进行补充说明。

3.4.3 非功能需求的说明要求

非功能需求涉及法律法规、应用程序标准、质量属性(可用性、可靠性、性能、兼容性、可移植性等),以及设计约束等方面的需求。在这些需求的描述方面,一定要注意使其可量度、可验证,否则就容易流于形式,形同摆设。可以通过质量场景图对系统需要满足的质量要求进行定性或定量的说明。

(1) 易用性:例如指出普通用户和高级用户要高效地执行某个特定操作所需的培训时间;指出典型任务的可评测任务次数;或者指出需要满足的易用性标准(如 IBM 的 CUA 标准、Microsoft 的 GUI 标准)。

(2) 可靠性:包括系统可用性(可用时间百分比);平均故障间隔时间(MTBF,通常表示为小时数,但也可表示为天数、月数或年数);平均修复时间(MTTR,系统在发生故障后可以暂停运行的时间);精确度(指出系统输出要求具备的分辨率和精确度);最高错误率或缺陷率(通常表示为 bugs/KLOC,即每千行代码的错误数目或 bugs/function-point,即每个功能点的错误数目);错误率或缺陷率(按照小错误、大错误和严重错误来分类,需求中必须对“严重”错误进行界定,例如,数据完全丢失或完全不能使用系统的某部分功能)。

(3) 性能:包括对事务的响应时间(平均、最长);吞吐量(例如每秒处理的事务数);容量(例如系统可以容纳的客户或事务数);降级模式(当系统以某种形式降级时可接受的运行模式);资源利用情况:内存、磁盘、通信带宽等。

阐述具体的应用领域对产品性能的需求,并解释它们的原理以帮助开发人员做出合理的设计选择。确定相互合作的用户数或者所支持的操作、响应时间以及与实时系统的时间关系。还可以在这里定义容量需求,例如存储器和磁盘空间的需求或者存储在数据库中表的最大行数。尽可能详细地确定性能需求,可能需要针对每个功能需求或特性分别陈述其性能需求,形成多个具体的性能场景。例如,“在运行微软 Windows 2003 Server 的 2.4GHz 的计算机上,当系统至少有 50% 的空闲资源时,95% 的数据库查询必须在 2s 内完成”。

(4) 安全性:包括系统安全、完整或与私人问题相关的需求,这些问题将会影响到产品的使用和产品所创建或使用的数据的保护,定义用户身份确认或授权,应明确产品必须满足的安全性或保密性策略。

3.5 需求说明书内容框架

需求变更是无法避免的,发生需求变更的一个重要原因是系统周围的世界在不断变化,从而要求系统适应这个变化。在项目生命周期的任何时候或者项目结束之后都可以有需求变更。与其希望变更不会来临,不如希望初始的需求在某种程度上做得很好而使得没有变更需求,最好是项目准备时想到对付这些变更,以防变更真的到来。但不管做多少准备和计划都不可能阻止变更,期望项目在需求冻结后再开始是很难做到的。

需求规格说明的主要内容,即需求分析应交付的主要文档是需求规格说明。软件需求规格说明的一般格式如下:

- (1) 引言;
- (2) 任务概述;
- (3) 数据描述;
- (4) 功能要求;
- (5) 质量需求;
- (6) 运行需求;
- (7) 其他要求;
- (8) 附录。

3.6 需求原型工具 Axure

Axure RP 是一个专业的快速原型设计工具,Axure 代表美国 Axure 公司,RP 则是 Rapid Prototyping(快速原型)的缩写。Axure RP 是美国 Axure Software Solution 公司旗舰产品,是一个专业的快速原型设计工具,让负责定义需求和规格、设计功能和界面的专家能够快速创建应用软件或 Web 网站的线框图、流程图、原型和规格说明文档。作为专业的原型设计工具,它能快速、高效地创建原型,同时支持多人协作设计和版本控制管理。

Axure RP 已被一些公司采用。Axure RP 的使用者主要包括商业分析师、需求分析

师、信息架构师、可用性专家、产品经理、IT 咨询师、用户体验设计师、交互设计师、界面设计师等,另外,架构师、程序开发工程师也在使用 Axure。

Axure 的可视化工作环境可以让用户轻松快捷地以鼠标单击的方式创建带有注释的线框图。不用编程,就可以在线框图上定义简单连接和高级交互。在线框图的基础上,可以自动生成 HTML(标准通用标记语言下的一个应用)原型和 Word 格式的规格。