

软件开发中的需求获取

赵艳玲

(珠海城市职业技术学院 广东 珠海 519000)

摘要:针对需求获取这一重要问题,提出了“需求陈述”和“构想文档”。“需求陈述”陈述了软件方案的需要,“构想文档”是为软件开发团队创建的,有助于开发人员明了项目的目的和范围。此外,对需求来源的确定、需求的收集方法进行了描述,对需求获取中常见的问题进行了讨论。

关键词:需求获取;需求陈述;构想文档;数据流图

中图分类号:TP311.5

文献标识码:B

文章编号:1004-373X(2007)11-184-02

Requirements Elicitation of Software Development

ZHAO Yanling

(Zhuhai City Polytechnic, Zhuhai, 519000, China)

Abstract: Requirements elicitation is an import problem. This paper proposes A Statement of "Requirements" and "Vision Documents". Vision Documents state the need of software, the Vision Documents is created for the software development team so that the team has clarity on the purpose and scope of the project. This paper confirms the source of requirement and explains method of collect requirement and answers the most common question in "Requirements Elicitation".

Keywords: requirements elicitation; a statement of requirements; a vision documents; data flow diagram

许多大型应用系统的失败,最后均归结到需求分析的问题。要么获取需求的方法不当,要么客户配合不好,导致客户对需求不确认。而需求分析的输出文档是《用户需求报告》,是客户、软件开发人员和项目管理人员三者必须遵守的一根基线,是供方交付产品和需方验收产品的依据。另外需求分析要占用整个软件开发时间或工作量的30%左右。需求获取中的错误,会在后续的设计和实现中进行发散式的传播。因此能正确获取需求就成为了项目开发的关键。从各种角度采取各种制度措施对需求正确理解,从而保证项目质量是现在软件开发的发展方向。

1 需求获取的过程

需求获取是需求分析的基础,正确、完整的获取需求是保证项目成败的关键。下面通过需求陈述、创建构想文档、确定需求来源、需求收集的方法这4个方面进行阐述。

1.1 需求陈述

需求工程是软件开发中管理需求的过程。需求工程创建的第一份文档是需求陈述,用于在项目开发之初理解客户的需求。需求陈述是一份文档,陈述软件方案的需要,并对可能的规格要求加以说明。并用来明确软件开发的用途,他不仅要说明软件有什么用,还要在宏观层面上明确软件应具备的特性。

通过需求陈述我们可以知道:为什么需要开发这个软

件;在这个项目里,哪些人是关键的利益相关者;这个项目预期要实现哪些主要功能。因此作为需求陈述的基本组成就应该有:公司描述,开发项目的必要性,对将要开发的软件描述,软件开发计划、风险。

1.2 创建构想文档

构想文档是为软件开发团队创建的,有助于开发人员明了项目的目的和范围。从宏观层面上显示软件的需求,为待开发项目提供一个结构清晰的概要,确保项目的计划、设计和开发等阶段能依次有序地展开。因此构想文档的组成应包括:公司描述,需求概要,项目目标,项目利益相关者,软件主要功能,项目约束。

1.3 确定需求来源

需求抽取是从相关来源为软件开发收集需求的过程。需求来源可以是公司员工和外部人员(如供应商等),或公司文件,包括手册、文书、表格和报表等,也包括现有应用系统的文档。

1.4 需求收集的方法

需求收集方法可以有背景阅读、文档取样、面谈、调查问卷、问题根本原因分析等5个方法。

在启动需求收集过程之前,充分了解委托公司及其业务流程非常重要。这可以通过大量的背景阅读来完成,可供阅读的背景信息包括:公司报告、公司图表、策略手册、岗位描述、报表、现有系统的文档。

文档取样指收集公司保存的各类文档。这些样本文档可用于决定软件系统的输入、输出,在存在现有软件系

统的情况下,这是个非常理想的方法。另外,为了理解系统输入/输出,在存在现有软件系统的情况下,这是个非常理想的方法。另外,为了理解系统输入/输出接口,还可以收集些屏幕快照。

面谈是使用最为广泛的一种需求收集方法。系统分析员可向管理层了解有关公司目标,再从员工处收集与业务流程相关的信息。面谈的优点是面对面的沟通能使系统分析员对需求更敏感,也更容易采纳被访者的意见。分析员能与被访者就他的工作性质进行更为深入的探讨,这种效果是其他需求抽取技术所不能企及的。面谈是一种耗时而且开销较高的需求抽取方法。如果被访者的意见有冲突,会使确定需求变得更困难。在访谈中要注意不要记录与议题无关的信息,问的问题必须与议题有关,问的问题必须与被访者的工作直接相关。

调查问卷是在项目开发中收集信息的手段之一。调查问卷包含一系列经过设计的问题,通常被认为是广泛收集数据的最经济的一种手段。如果应用系统的用户是一般大众,或者分析员觉得有必要了解系统的具体用法,那么调查问卷这种方式是非常有效的。调查问卷一般包括是非题和问答题。调查问卷的优点:调查问卷是一种最经济的广泛收集数据的方法,一个精心设计的调查问卷很容易进行需求分析。调查问卷的缺点:调查问卷不容易设计得好。不能对特定问题进行深入的探究,不象在电话调查或面对面的沟通中,必要的话,可对一个问题追根到底。

问题根本原因分析是分析系统存在的问题,并找出其根源及解决办法,以消除故障因素。可进行原因分析的技术有:再看一遍需求,列出所有类似的线索,查找错误发生的机会,鱼骨图排列分析。这些技术不一定能把原因马上找到,但能拓宽分析人员的思路,利于更进一步的仔细检查,与根本原因很接近。

2 需求获取中常见的问题分析及对策

(1) 最常见的问题是不同的人对“需求”一词的不同理解。

例如,项目负责人理解的“需求”可能是软件产品在宏观层面的功能,而开发人员理解的“需求”可能是指用户界面的设计。

原因在于客户提供的需求往往只是一些粗略的想法,项目的利益相关者在谈到“需求”时缺乏准确的描述,从而造成开发团队对需求到底应该多详细期望也不一样。

解决这个错误的第一步是要理解需求有不同的类型,所有的需求都是必须了解的。第二步是要让开发小组成员了解需求工程的概念和术语。

(2) 客户参与得不够,或缺乏与客户的沟通。

原因是一方面客户往往想当然地以为开发人员已经知道需求了,另一方面,开发人员则可能在没有足够了解

的情况下就擅自做出很多需求决定。结果,最终产品很可能因为不符合客户需求而被拒。

有效解决这一问题的方法之一是要确定客户端有多少类型的用户组,每个用户组要使用软件的哪些功能,有什么样的访问权限。另一种有效的办法是在每个用户组里认定一个代表,由这个代表来收集和提供组员的需求。这些代表还可参与评估开发原型,评审软件需求说明,检查需求的完整性和正确性。

(3) 需求陈述的模棱两可。

原因在于不同的人对需求有不同的解读方式,每个人都觉得自己的理解是正确的,需求就这样含糊糊而没有发觉。等到发现时,往往要花很大的代价才能解决。

解决方法是在编写需求说明时避免使用太主观或者含义模糊的字眼。例如对“健壮的”、“用户友好的”、“直觉的”等用词每个人都可以有不同的解释,因此必须避免。需求用词要求含义尽可能清晰、准确。验证需求的一个好办法是制定测试计划,为系统建模,再将两者都交由客户评审。

(4) 需求的优先次序不正确。

原因是在很多的案例中,往往有超过90%的需求都被认为是高优先级的。将所有需求都看成是重要的,这种做法会让项目经理很难回应新需求或更改项目进度,最后也难以保证产品质量。

解决方法是依据客户的重要性来分析需求的优先级,另外,还要考虑到开销和相关的技术风险。

(5) 为了使系统看起来更好,在产品的某些用得极少的功能上花费大量的时间和精力。

原因是很多时候,客户不大区分哪些功能是重要的、用得多的,哪些功能很少用。结果,开发人员为了实现那些几乎用不上的功能浪费了大量宝贵时间。

解决方法是要对每个问题追根溯源,例如,追究需求的源头是高层系统需求、业务规则还是行业标准。另外,还要确定每个功能的受益人是谁。

(6) 只注重了功能方面需求,而没注意性能方面需求。使项目不能经受各种信息量的考验。

原因是分析员把大部分时间都花在了收集与项目操作有关的需求上,对系统性能估计不足。

解决方法是需求分析和技术实现不能只侧重于系统的功能性,还要考虑到一些潜在的需求。必须定义系统的范围,这个范围要涵盖系统所有方面的需求以及系统可变需求。

3 创建业务流程模型

业务过程建模是一种方法,用来理解公司的业务过程及相关要素,如数据流、工作流和主要任务等。

(下转第188页)

FLASH 进行读写命令/数据操作。如果此程序工作在 OS 上的 application 层的话,MMU 已经屏蔽程序对底层硬件的直接访问,所以需要 MMU 进行设定,为 NAND FLASH 开辟一块 Memory 映像区域,这样就可以通过 OS 对底层的 NAND FLASH 进行操作。以该系统为例,使用 CPU 的 CS1 引脚控制 NAND FLASH 的 CE 信号,先将其映像为 0x24000000 地址,此时,对 0x24000000 地址读写即对 NAND FLASH 芯片进行数据读写,而对 0x24000002 地址写数据,使 CPU 的 A1 地址引脚为高,即对 NAND FLASH 发送 command 命令,同样,对 0x24000004 地址写数据,即对 NAND FLASH 发送 address 数据。

在对 NAND FLASH 发送命令/数据之后,由于程序运行速度比 FLASH 芯片快很多,需要在每一次操作之后插入若干等待周期,并利用 CPU 的 GPIO 检测芯片 R/B 信号。直至芯片完成本次操作再进行下一步操作。

需要注意的是,在对 FLASH 发送命令数据过程中的等待,没有反馈信号可以检测,只能通过反复调试确定其所需等待时间。

在设计中采用 CPU 的 CS1 信号对 NAND FLASH 进行 CE(片选)控制。此处不能采用 CPU 的 GPIO 进行控制,因为在嵌入式设备的 ARM CPU 中,CPU 本身采用了指令、数据自动预读的高速缓存技术和流水线技术。因此,当程序在 NOR FLASH 里面直接运行的时候(目前绝大多数嵌入式系统采用的方式),在运行任何两段相连的代码中间,CPU 都有可能对 NOR FLASH 进行指令或数据的预读操作,从而产生大量的 RE,OE 信号和地址信号。如果使用 GPIO 控制 NAND FLASH 的 CE 信号则无法避

免这种影响。CPU 的 CS1 信号是由 CPU 内部自动产生,因此在 CPU 预读期间,CS1 信号可以有效屏蔽 NAND FLASH 芯片。并且,由于 NAND FLASH 芯片支持 CE don't care 模式,在 CE 无效的情况下,芯片本身的工作状态并不会被干扰,由此保证了 NOR FLASH 和 NAND FLASH 在同一 CPU 界面中互不干扰的稳定运行。对于 CS1 信号的宽度等参数,也需要在实验中进行调节,才能保证整个系统快速稳定的运行。

4 NAND FLASH 在系统中的读写速度

经过测试在该系统平台中,OS 为 Palm OS 5.4;CPU 使用 PXA270 312 MHz;SDRAM 使用 Samsung 的 16 b data width HYB25L256160AF - 7.5 @ 104 MHz;NAND FLASH 选用 Samsung 128 MB 8 b I/O NAND FLASH K9F1G08U0A 达到在文件系统下面的读/写的速度为 3 MB/s,擦除的速度为 65 MB/s,在手持式设备中运用性能已经够了。

参 考 文 献

- [1] Intel Company. Intel® PXA27x Processor Family Developer's Manual.
- [2] Intel Company. Intel® PXA27x Processor Family Design Guide.
- [3] Samsung Company. Samsung NAND FLASH K9F1G08U0A Spec.
- [4] 袁文菊,孙天泽,付晓江.大容量 NAND FLASH TC58DVG0ZAIPT00 在嵌入式系统中的应用[J].国外电子元器件,2004(12):63-65.

(上接第 185 页)

建模可以理解客户的业务过程以及需求,理解业务过程有助于明了和理解软件需求;建模可以确定业务过程中的各要素,如子过程和数据流,有助于确定信息流和软件各组成部分;建模可以与客户一起验证业务过程,使用业务过程模型来验证业务过程有助于在开发小组和客户之间建立共识;建模可以明确哪些任务不能自动化,分析业务过程有助于定义自动化边界,确定哪些任务是不能自动完成的,哪些是技术、资金不允许的。

可用于创建业务过程模型的工具具有:数据流图、用例图、流程图等。其中数据流图显示了数据是如何在系统中产生、传输、加工处理、使用、存储的过程。

4 结 语

众所周知,需求调研不充分,用户需求描述不完整、不准确,轻则影响项目建设的顺利程度,重则影响应用系统

的质量,甚至决定项目的成败。需求获取就是在只有一个大体需要或只是一个概念上的想法的基础上,软件开发人员通过广泛的调查,并从服务意识角度出发,带领客户或引导客户一起“搞清楚”需求。而调查是广泛的,包括:业务专家、一线人员、市场情况、竞争对手等。调查是深入的,如:问卷、访谈、研讨会、现场了解、收集资料及与业务人员交朋友。

参 考 文 献

- [1] 梅宏.软件工程(实践者的研究方法)[M].北京:电子工业出版社,2003.
- [2] 张海藩.软件工程[M].2版.北京:人民邮电出版社,2006.
- [3] [美]莱芬韦尔.软件需求管理用例方法[M].2版.北京:机械工业出版社,2004.