

3.1 系统需求的来源

系统需求应该从所属组织的业务用例中进行获取。

业务用例是组织存在的价值,不是经常发生变化的,经常改变的是业务用例在组织内部实现的流程,也就是说,组织引入系统后,改善了业务流程。

例如,银行的业务用例包括存款、取款、转账等,这些业务用例自银行出现就存在了,多少年都没有改变,改变的只是这些业务用例的实现流程,从这些业务用例的序列图的变化就会发现业务流程从纯人工变为自动化实现,即组织使用某些系统替换了原来的业务工人。如银行使用点钞机替换了点钞技能强的员工,使用自动取款机、手机银行替换了更多的柜员和柜员系统。因此,对于银行的存款和转账业务用例来说,银行对每个用例都提供多种实现,例如,柜台方式取款见图 3-1、ATM 设备取款见图 3-2,柜台方式转账见图 3-3、手机银行方式转账见图 3-4。

可见,组织中引入系统,改变了相关业务用例的实现流程,而流程的改变为银行降低了用人成本,提高了准确率,扩大了顾客群体。

组织中引入系统,系统通常在以下三方面发挥作用,改善相关业务用例的实现流程^[1]。

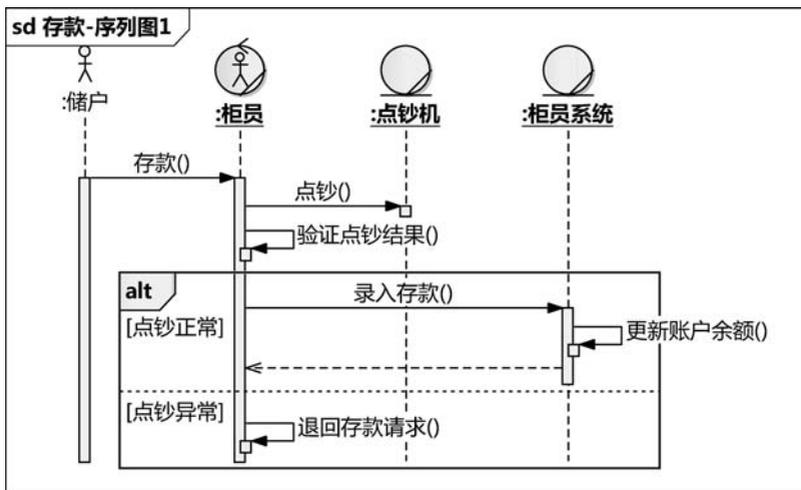


图 3-1 存款用例的序列图——使用柜员系统/点钞机

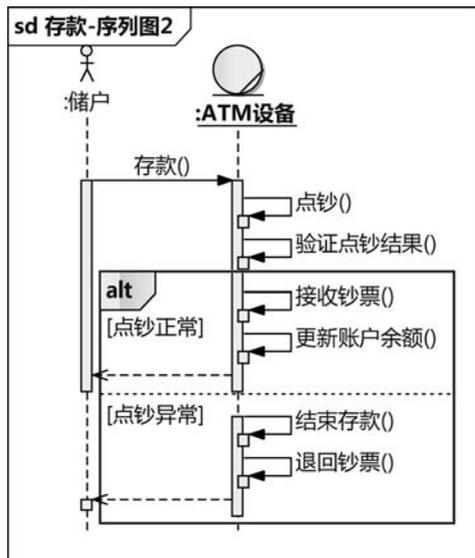


图 3-2 存款用例的序列图——使用 ATM 设备

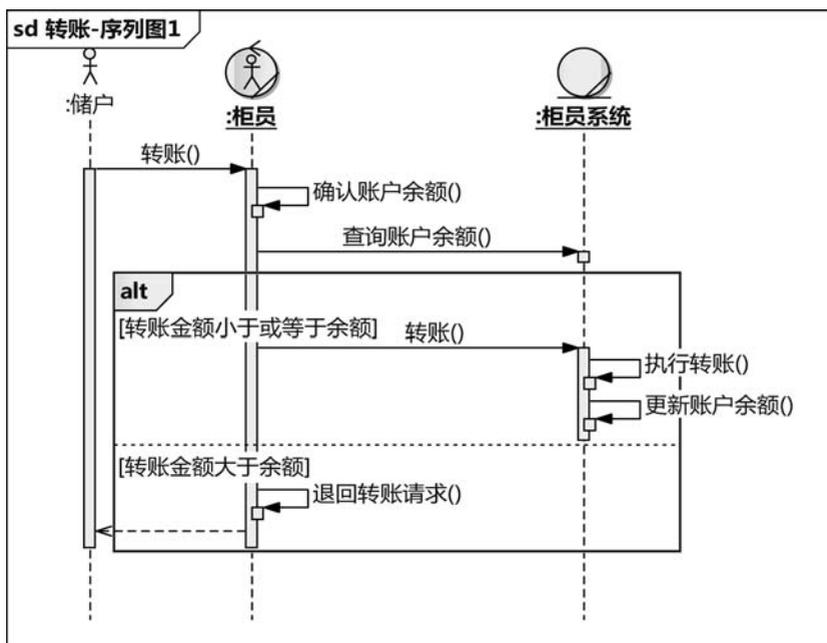


图 3-3 转账用例的序列图——使用柜员系统

(1) 系统将原物流改为信息流。例如,某组织以前的办事流程主要靠纸质在办事人员之间进行流转,引入某业务系统后,在不同办事人员之间流转的是系统中的信息。

(2) 系统改善了信息流转。例如,医院原来的看病流程是:①患者在挂号窗口挂号,等待挂号系统叫号;②医生开具处方后,患者去收费窗口,划价人员将药品信息录入财务系统,划价收费;③之后患者去取药窗口,药品管理系统记录取药品种和数量。医院引进新的系统后,看病流程变为:①患者在挂号窗口挂号,系统录入医保卡或银行卡信息,等待系统

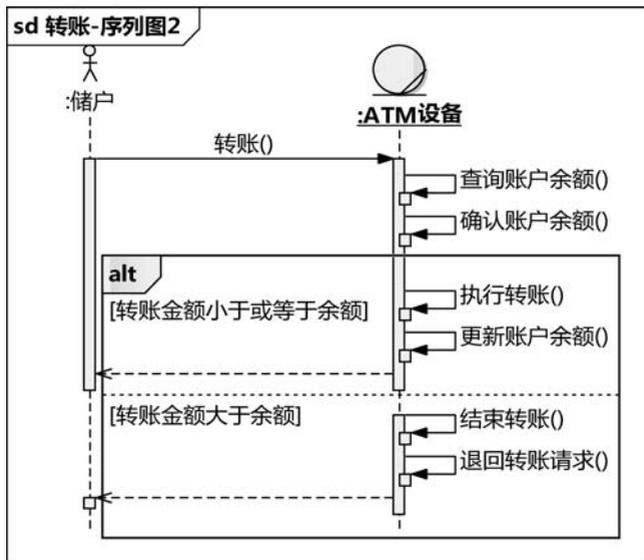


图 3-4 转账用例的序列图——使用手机银行

叫号；②医生使用系统开具处方后，系统直接扣除药费，并记录取药品种和数量；③患者去取药窗口直接取药。可见，原来医院的看病流程中至少有三个独立系统，信息不能在三个系统中直接流转，引进新的系统后，信息流在一个系统内部流转，为患者看病节约了等待时间。

(3) 系统封装了领域逻辑。例如，银行引进点钞机，将人工点钞技能变为系统实现，无须再花费人力物力培养柜员的点钞技能；出现手机银行后，手机银行封装了转账的业务逻辑，替代了柜员和柜员系统。如果作战部队引入智能指挥信息系统，封装优秀指挥人员的指挥决策技能，就能减少指挥决策失误的概率。

可见，要获取系统需求，就要从研究(系统所属)组织的业务用例开始。系统用例来自组织的业务用例。

3.2 系统是组织的部件

系统作为组织中的业务实体，和业务工人一样，属于组织内部的部件。在一个组织中，通过对业务工人和业务实体(系统)之间的协作关系进行良好设计，从而对外部涉众提供有价值的服务。服务的优劣受到业务工人素质的影响，同样也受到系统能力的影响。

系统和业务工人一样都属于组织的部件，系统可以被业务工人使用，也可以直接被外部涉众使用。也就是说，系统用例的主执行者可以是业务工人，也可以是组织的外部涉众。例如某车载侦察系统，其主执行者有业务工人(侦察员)，也有外部涉众(上级指挥人员)。

3.3 分析方法综述

系统作为所属组织的部件，是为提升组织的业务价值而存在的，系统是为组织更好地实现业务用例而研制，所以系统的能力需求来自组织的业务用例(即组织的业务价值)。

系统需求分析的第一步首先要找对组织,将组织作为研究对象,发现组织的业务用例,并确定现行状态下,组织的这些业务用例是如何在组织内部(各部门和系统)分工协作完成的。之后,将待采购的新系统或待改进的系统放在组织内部,确定系统能够在哪些方面改进哪些业务用例的实现过程。

上述工作完成后,按照 GJB 2786A 的要求,可以将工作内容记录在《运行方案说明》中。

之后,将系统作为研究对象,对得到的系统用例(能力需求)进行详细分析,描述每个用例的规格。相应的工作内容记录在《系统/子系统规格说明》中的第 3 章的 3.2 系统能力需求中。

系统需求分析的第二步工作是基于得到的系统用例,进行外部接口分析,确定系统的外部接口的物理形式、传输速率,以及与软件相关的数据协议。相应的工作内容记录在《系统/子系统规格说明》中的第 3 章的 3.3 系统外部接口中。

系统需求分析的第三步工作是确定系统的内部接口。此处的内部接口是指系统的子系统之间的接口,即当在系统需求分析阶段,鉴于业务类型或业务职责,可以显式将系统划分为若干子系统时,需要明确子系统之间的接口需求。相应的工作内容记录在《系统/子系统规格说明》中的第 3 章的 3.4 系统内部接口中。

系统需求分析的第四步工作是确定系统内部数据需求。内部数据需求是指与功能需求相关的数据需求,即找到问题域中存在的业务实体,确定它们之间的逻辑关系、数量关系和结构规则。相应的工作内容记录在《系统/子系统规格说明》中的第 3 章的 3.5 系统内部数据需求中。

系统需求分析的第五步工作是分析系统的质量因素。按照相关国标要求,主要对 6 类 21 项质量特性进行分析。相应的工作内容记录在《系统/子系统规格说明》中的第 3 章的 3.6 适应性需求、3.7 安全性需求和 3.11 系统质量因素中。

系统需求分析的第六步工作是进行设计和构造约束分析。这类约束主要包括四类:业务环境约束、使用环境约束、构建环境约束、技术环境约束。相应的工作内容记录在《系统/子系统规格说明》中的第 3 章的 3.12 设计和构造约束中。

3.4 分析之第一步——系统能力需求分析

3.4.1 确定组织

此处的组织是指需要采购新系统的或需要改进原有系统的组织。组织可大可小,重要的是所找组织的业务用例必须是与系统密切相关的,系统引进到该组织后,能够改善该组织的一些或全部业务流程。如果系统不能改善组织的业务流程,就一定没有找对组织。简而言之,新系统将成为所属组织的部件。

例如,某公司需要采购一套财务管理系统,目标是在确保账务正确性的基础上提高财务部门工作效率。那么,该系统对应的所属组织就应该是财务部门,而不是公司其他部门,或公司本身。

3.4.2 发现组织的业务用例

找到组织之后,要从组织的外部视角,发现组织对外提供的价值,这些对外提供的价值

就是组织的业务用例。

可见,业务用例是组织的用例,不是系统的用例,因此,业务用例不是思考系统提供什么“功能”,而是思考组织购买了这个系统,对组织价值的提升是否有帮助。思考的焦点是“执行者对组织的期望和组织对执行者的承诺”的平衡点。

业务用例代表组织的本质价值,稳定性高,很难变化,往往发生改变的是业务用例的实现,即业务流程发生变化。

例如,对于快递公司而言,其组织的业务用例见图 3-5。

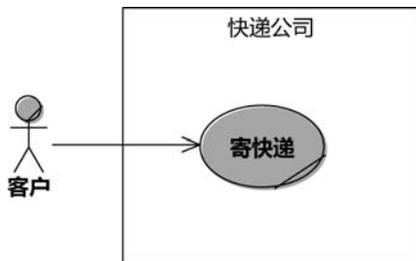


图 3-5 快递公司的业务用例

对于银行而言,其组织的业务用例见图 3-6。

对于电商机构而言,其组织的业务用例见图 3-7。

可见,这些组织的业务用例并不随着技术的发展 and 时间的流逝而改变,只是这些业务用例的实现方式可能随着技术更新而变化。

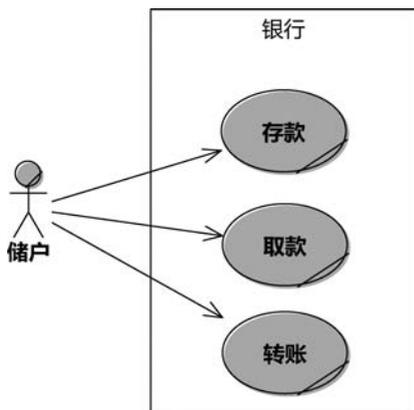


图 3-6 银行的业务用例

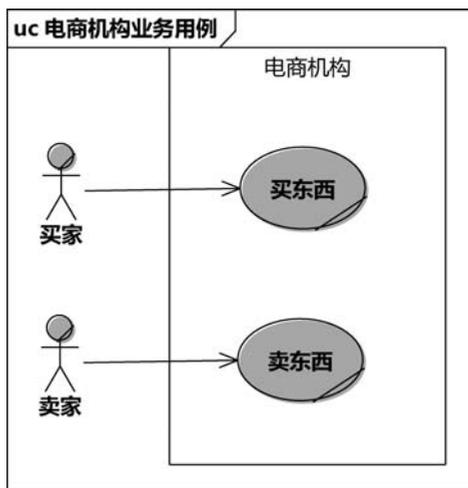


图 3-7 电商机构的业务用例

快递公司内部包含若干小组织,如人力资源部、财务部、综合管理部、快递业务部等,这些部门之间各类角色人员协作完成了快递公司对外提供的价值(寄快递)。

如果快递公司希望改善财务部的运行效率,那么就应该以财务部为研究对象,找出财务部的业务用例。财务部为组织内部的其他部门人员服务,包括员工报账、员工借款、员工工资发放、支出合同付款等,组织的业务用例见图 3-8。

3.4.3 确定系统用例

系统用例是表述系统第一类需求(功能需求)的方法。

系统是组织的(零)部件,在组织的内部,与组织内的其他(零)部件(业务工人和其他系统)一起协作,完成组织的业务用例。

确定系统用例的方法是使用组织的业务用例的序列图,在各个业务用例的序列图中分

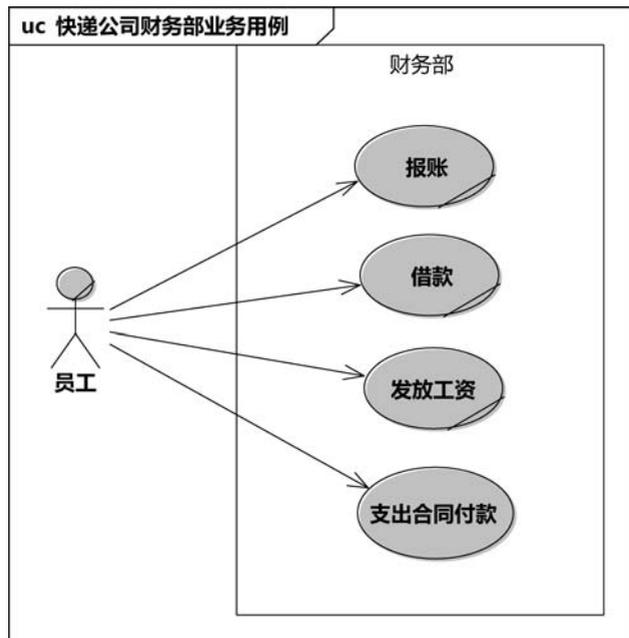


图 3-8 快递公司财务部的业务用例图

别找出系统的用例。

这个方法的基本原理为：业务用例的序列图表达了组织的部件之间如何协作实现业务用例,这种协作就是明确部件之间的职责,而职责就是能力需求。从这里也可以看出,能力需求是部件对外承担的责任,不涉及部件如何实现承担的责任。

具体的方法是,找出待改进的业务用例,使用序列图描述这个业务用例目前在组织内部的实现流程,之后将待改进系统或新研系统放在序列图中,对这个系统分配职责,即可得到系统用例。

在系统需求分析时,开发人员经常混淆组织的业务用例和系统用例,将组织的业务用例作为系统用例。

如图 3-9 所示,将“作战指挥”当作系统的一个大用例,下面分出第二层次的系统用例,更有甚者,还能再分出第三层次的系统用例。这类错误的根本原因就是组织的业务用例当作系统用例,系统用例虽然来自组织的业务流程,但不是组织的业务用例的直接分解关系,而是在组织业务用例的实现过程中,系统作为组织的部件,被分配了一定的职责,这些职责才是系统用例。所以组织的业务用例的业务执行者和系统用例的主执行者是不相同的。

这类错误非常好辨识,因为在文档化表达时,这类错误的表现形式是把上一层用例当作一级目录,所以在能力需求目录下会出现多级目录,每个目录下又出现一个用例图。而实际上能够以目录结构出现的,只能是 UML 的包。也就是说,将一些用例组织在一个包中时,可以用用例包名作为目录出现,而不能是被分层的用例。

按照以下原则,从组织的业务用例的序列图中获取系统用例。

- (1) 指向系统的消息即是系统的职责,即系统用例。
- (2) 消息的发起者即是系统用例的主执行者。

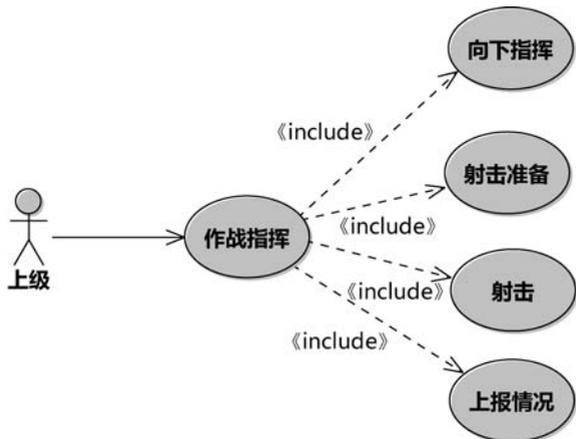


图 3-9 将组织业务用例当作系统用例的典型错误

(3) 从系统发起的消息指向的外部系统即是用例的辅助执行者。

其中,系统执行者是指在系统的责任边界之外,与系统做有意义交互的系统(含其他系统、人、时间),包括主执行者、辅助执行者。主执行者是发起系统用例的执行者,辅助执行者是为完成用例必须存在的,否则用例无法完成。

3.4.4 描述系统用例规格

建议用表 3-1 模板描述系统用例规格。

表 3-1 系统用例规格

用例名称	动宾结构命名	项目唯一标识符	UC-CSCI-001
研制要求章节			
简要描述	用例目标简述: 概述用例对外提供价值		
参与者	主执行者: 必须有主执行者 辅助执行者: 可以没有辅助执行者		
前置条件	该用例开始前,系统需要满足的约束。且是系统能够检测到的		
主流程 (代表用例核心价值的路径)	步骤	描述	
	1	第一步的主语是主执行者	
	2	(1) 一定聚焦于系统与外部的交互过程 (2) 不要没有主语,且主语只能是主执行者或系统 (3) 使用主动语句,突出主语承担的责任 (4) 使用系统所属组织的领域(核心域)的词语 (5) 不要描述交互过程中设计的细节 (6) 不要写系统不能负责的事情	
扩展流程	1a	对应基本流程中某个步骤中,系统需要处理的意外和分支。 注意: 扩展一定是动作选项,不能是数据选项。因为扩展将改变用例的基本流程,选择不同的数据并不能改变用例的流程,就不是扩展	
	1a1	针对 1a 发生的意外,系统的处理流程	
子流程	1a	对应基本流程中多次重复的一组步骤集合	
后置条件	该用例结束后,系统需要满足的约束,且是系统能够检测到的		
规则与约束	业务规则、数据约束、性能需求等		

编写系统用例需求规格说明需要掌握的要点^[1]：

- (1) 用例的命名规则,使用动宾结构。
- (2) 第一个步骤的主语应该用用例的主执行者。
- (3) 前置/后置条件必须是系统能够检测到的。
- (4) 基本流程是主执行者最想看到的、最关心的路径。
- (5) 流程步骤要使用主动语句,主语只能是执行者或系统。
- (6) 步骤聚焦于交互的输入输出,突出交互的目的,所以不要将步骤写得零散,要把交互的目的从交互的细节中分离出来。
- (7) 每个步骤必须遵循可理解可验证原则。
- (8) 步骤中不要描述系统不能负责的事情。
- (9) 步骤中不要描述设计内容。对于这条,存在一个普遍性的错误认识:“写得细的是好需求”。但是开发人员写得“细”不是需求(问题域)的细,而是设计(解决方案)的细。
- (10) 尽量不要涉及界面组件;因为界面组件通常不是需求。
- (11) 因为辅助执行者是被系统激励后才开始执行自己的职责,所以建议写“系统请求××做某事”,不能写“××做某事”,以此表示系统对外部发生的交互。
- (12) 步骤可以循环。
- (13) 步骤中不能出现“如果”。“如果”应由扩展流程描述。
- (14) 扩展是系统要处理的意外和分支,是系统能感知和要处理的,一般出现在执行者的选择和系统验证处。
- (15) 规则与约束:包括数据约束、业务规则和性能要求等。
- (16) 数据约束建议用表达式表述字段列表;此处的字段是指有物理含义的数据,不要过早关注数据细节,如定义某个物理数据字段的字节位置等。
- (17) 业务规则,包括事实、推理、业务约束。

例 1: ATM 系统的取款用例,用例规约见表 3-2。

表 3-2 ATM 系统的取款用例规约

用例名称	取款	项目唯一标识符	UC-ATM-001
研制要求章节			
简要描述	储户使用银行卡,可通过 ATM 设备提取现金		
参与者	主执行者: 储户 辅助执行者: 银行后台系统		
前置条件	ATM 设备处于准备就绪状态;吐钞口关闭,与银行后台系统连接正常		
主流程	步骤	描 述	
	1	储户插入银行卡,ATM 设备从银行卡磁条读取账户信息,验证银行卡合法性	
	2	储户输入密码,ATM 设备要求银行后台系统验证账户和密码正确性	
	3	储户提交取款金额,ATM 设备检查取款金额有效性	
	4	ATM 设备要求银行后台系统验证账户、取款金额有效性	
	5	ATM 设备检查预留现金是否支持取款金额	
	6	ATM 设备打开出钞口,出钞,提示取钞	
	7	ATM 设备实时检查钞票出钞状态(是否被取走),钞票被取走后,关闭出钞口	
	8	ATM 设备要求银行后台更新账户信息,记录日志	
	9	ATM 设备提示储户是否继续取款	
	10	循环 2~9 步骤	

续表

扩展流程 1	1a	银行卡无效
	1a1	ATM 设备提示无效银行卡,退出卡片。用例结束
扩展流程 2	2a	密码错误
	2a1	ATM 设备提示密码输入错误次数
	2a1-a	达到最大次数
	2a1-a1	ATM 设备提示密码输入次数达到最大次数,吞卡。用例结束
	2a2	回到主流程 2
扩展流程 3	3a	取款金额不是 100 的整数倍
	3a1	ATM 设备提示取款金额不是 100 的整数倍
	3a2	回到主流程 3
	3b	单笔取款金额超限
	3b1	ATM 设备提示单笔取款金额超限
	3b2	回到主流程 3
扩展流程 4	4a	卡内余额不足
	4a1	ATM 设备提示卡内余额不足
	4a2	回到主流程 3
	4b	达到每日最大取款限额
	4b1	ATM 设备提示达到每日最大取款限额,退出卡片,用例结束
扩展流程 5	5a	ATM 设备现金不足
	5a1	ATM 设备提示现金不足
	5a2	回到主流程 3
扩展流程 6	6a	出钞失败
	6a1	ATM 设备提示设备故障,退出卡片
	6a2	ATM 设备记录故障信息,用例结束
扩展流程 7	7a	钞票超时未被取走
	7a1	ATM 设备关闭出钞口,提示取款失败,退出卡片
	7a2	ATM 设备记录取款失败信息,用例结束
扩展流程 8	8a	储户选择退出
	8a-1	ATM 设备退出卡片,用例结束
后置条件	ATM 设备恢复到就绪状态	
规则与约束	(1) 密码为 6 位数字 (2) 密码输入错误 5 次,ATM 设备吞卡 (3) 每笔取款金额最多 5000 元 (4) 每日最多取款金额 20 000 元 (5) 与银行后台系统交互的数据 (5-1) 主流程第 2 步,账户+密码 (5-2) 主流程第 4 步,账户+取款金额 (5-3) 主流程第 8 步,取款成功+账户+取款金额	

例 2: 一个自动售饮料机出售三种饮料: 可乐,雪碧或红茶,顾客投入一元五角硬币,选择饮料品种,机器送出相应饮料。若投入两元硬币,机器送出饮料同时退还五角硬币。

自动售饮料机的系统用例——售卖饮料的用例规约见表 3-3。

表 3-3 自动售饮料机的系统用例——售卖饮料用例规约

用例名称	售卖饮料	项目唯一标识符	UC-YLJ-001
研制要求章节			
简要描述	顾客向系统投入硬币,选择饮料种类后,系统推出顾客选择的饮料和找零的硬币		
参与者	主执行者: 顾客 辅助执行者: 无		
前置条件	系统处于就绪状态: 投币口打开		
主流程	步骤	描 述	
	1	顾客投入硬币	
	2	系统开始计时,检查硬币的有效性,记录投币次数	
	3	系统关闭投币口,提示选择饮料种类,并开始计时	
	4	顾客选择饮料种类	
	5	系统检查顾客所选择饮料的存量	
	6	系统计算找零金额,推出饮料	
扩展流程 1	7	系统打开投币口,记录售卖的饮料种类	
	2a	硬币数额不足	
	2a1	系统提示硬币数额不足,继续投币	
	2a2	回到主流程 1	
	2a1-a	顾客取消购买,进入子流程 Z1	
	2b	硬币非法	
	2b1	系统提示硬币不合法,退出已投入的硬币,用例结束	
	2c	硬币大于 1.5 元,但是系统无零钱	
	2c1	系统提示并退出已投入硬币,用例结束	
	2d	T1 时间内,顾客投入硬币次数不够	
2d1	系统提示超时,并退出已投入硬币,用例结束		
扩展流程 2	3a	顾客取消购买,进入子流程 Z1	
	3b	T2 时间内,顾客未选择饮料	
	3b1	系统提示等待时间超长,退出已投入的硬币,用例结束	
扩展流程 3	5a	顾客选择的饮料已售空	
	5a1	系统提示所选饮料已售空,请选择其他饮料	
	5a2	返回主流程 3	
扩展流程 4	6a	需要找零	
	6a-1	系统推出找零硬币	
子流程	Z1	顾客取消购买	
	Z1-1	系统退出已投入硬币,用例结束	
后置条件	系统恢复到就绪状态: 投币口打开		
规则与约束	(1) 饮料种类包括: 可乐、雪碧、红茶,单价 1.5 元 (2) 系统只接受 1 元、5 角硬币 (3) 顾客一次只能购买一瓶饮料 (4) 一次只能投一个币 (5) 一次购买,5 角硬币最多投币三次; 其余投币两次		

3.5 分析之第二步——系统外部接口分析

系统的外部接口是指系统与系统外部的其他硬件和软件之间的接口。此处的接口是指两个接口实体(软件和硬件)对象之间形成连接关系的物理形态和数据交互协议。

首先,确定系统外部接口图。系统的外部接口图来自系统用例图,即系统用例图中的主执行者和辅助执行者一定是系统的外部接口图中对应的外部实体对象。

要避免出现以下几类问题。

(1) 系统外部接口图中出现的均应该是物理实体,不能在接口的两端出现“某接口”“某数据”“某信息”之类的非实体对象。

(2) 外部接口图中要如实体现接口的物理形态,如以太网、串口、IO口等,且数量要与实际一致,不要将多路同物理形态的接口合并为一个。

之后,根据外部接口图给出全部外部接口的概述,见表 3-4。

表 3-4 系统外部接口描述

序号	接口名称	标识	接口类型	接口用途	外部实体名称	外部实体状态
			网络、串口、CAN总线、IO等			新研、改进、复用、货架

其中,接口用途需要描述该接口上交互的数据,以及交互数据的目的。需要注意的是,在一个接口上需要交互不同的数据,以支持不同的系统用例,所以此处需要分开详细描述。

最后,确定每个接口的规格,见表 3-5。

表 3-5 某接口规格说明

接口名称	某接口		接口标识	JK-OU-001
接口实体	接口两端的软件或硬件		接口类型	网络等
接口用途	简述接口两端的软件或硬件使用该接口的用途			
接口数据	(1) 说明该接口上传输数据的种类 (2) 说明该接口上传输的数据的编码格式:如果每类数据使用相同的编码格式,用其他的表说明该格式;否则,逐一说明每类数据使用的编码格式;如果编码格式为标准格式,说明即可 (3) 说明每类数据的具体格式定义。如果数据较多,可以在附录中或使用 GJB 438B 的《接口设计说明》模板详细说明			
接口通信特征	通信链路	如网络的带宽、串口速率等		
	数据传输	非周期性/周期性、单向或双向传输		
接口协议特征	说明接口是否有同步机制,详细说明同步机制是如何设计的 接口上的每类数据的传输层协议等			

3.6 分析之第三步——系统内部接口分析

系统内部接口是指组成系统的部件之间的接口。在系统需求分析活动中,系统应该被当作黑盒对待。此时,系统的内部组成部件还没有被系统架构设计师设计出来,其内

部接口无法明确。但是,若系统需求分析时,已经明确了组成系统的各个子系统时,此时的系统内部接口就是指各个子系统之间的接口,参照系统外部接口分析的内容进行分析即可。

3.7 分析之第四步——系统内部数据需求

系统内部数据需求是指从系统用例中提取的相关数据需求,用 UML 的类表示就是指实体类,即系统用例的基本流程实现中的主要承载实体。

找到这些类,使用 UML 类图描述这些类所代表的事物之间的关系,就是系统内部数据需求分析所做的工作。内部数据需求分析的目标是找到“问题域”中存在的实体类,确定它们之间的逻辑关系、数量关系和结构规则。

一个用例可以有多个实体类参与,一个实体类也可以参与多个用例。

此时的类图关注的重点是数据需求,即只关心类和类的属性,不关心类的操作。

识别实体类及其属性需注意以下要点。

(1) 类所表示的概念一定能够表示某类对象,且有取不同值的独立属性;即其概念是一个数据容器。

(2) 类的命名要用名词。

(3) 类命名要使用领域词汇;不能出现“与”“或”以及“表”“信息”“数据”。

(4) 类的属性直接描述类的特征。即属性应该表述为“类的什么”,不能表述为“类的什么的什么”。

例如,某电商组织在线销售标准配置的以及自定义配置的计算机,线上采购计算机的业务流程如下:买家选择货品付款下单后,系统将订单发送给厂家,厂家按照订单生产后,连同发票一起通过快递公司发货,快递公司发货管理人员将发货单录入电商平台供买家查询。收货人确认签收后,快递公司送货人员变更送货单状态为完成。

对这个业务流程进行分析,识别出电商平台的系统用例:买家一下单、厂家一发货、快递公司一更新快递信息;买家一查询订单、快递公司一买家签收,见图 3-10。

从上述用例规约(限于篇幅,上文未描述)中,找到以下能够表示数据容器的名词。

(1) 计算机。包括(厂家)标准配置和(买家)自定义配置计算机。

(2) 买家。买家需要提供的信息:姓名、地址、付款账号。

(3) 厂家。厂家需要提供的信息:名称、地址。

(4) 订单。订单信息:下单日期、收货人、收货地址、电话、货品编号、数量、价格、状态(未发货/已发货/已签收)。

(5) 快递公司。需要提供的信息:名称、地址。

(6) 送货单。送货单信息:派送人、派送人员电话、送货单号、收货人、收货地址、电话、货品编号、数量。

(7) 发票。发票信息:名称、地址、纳税识别号、货品名称、数量、价格等。

(8) 收货人。收货人信息:姓名、收货地址、电话。

这些类之间的关系见表 3-6,类图见图 3-11。

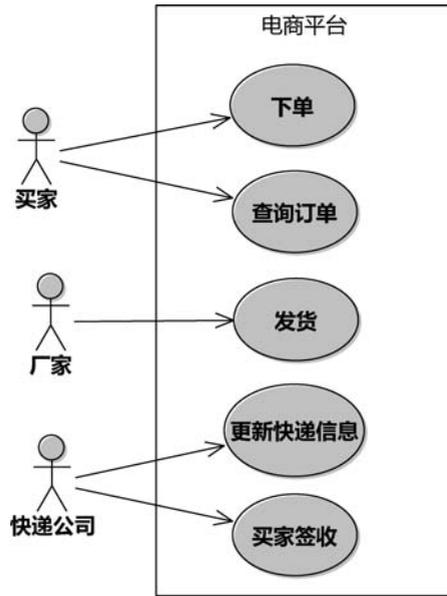


图 3-10 电商平台的用例图(局部)

表 3-6 类之间的关系

类 1	类 2	关系分析
买家	订单	1 个买家可以有 1 个或多个订单
订单	送货单	1 个订单只有 1 个送货单
订单	发票	1 个订单只有 1 个发票
订单	计算机产品	1 个订单包含至少 1 个以上计算机
订单	厂家	1 个订单仅属于 1 个厂家,但是 1 个厂家有多个订单
订单	收货人	1 个订单只有 1 个收货人
送货单	快递公司	1 个送货单仅属于 1 个快递公司,但是 1 个快递公司有多数送货单
送货单	厂家	1 个送货单仅属于 1 个快递公司,但是 1 个快递公司有多数送货单
厂家	计算机产品	1 个厂家有至少 1 种以上计算机产品

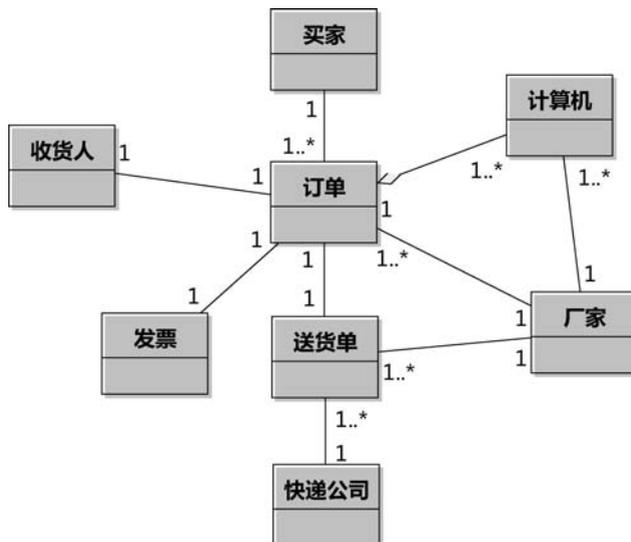


图 3-11 电商平台的领域类图(局部)

类图是面向对象系统的灵魂。这种方法适用于面向对象编程的软件,同样也适用于非面向对象编程的软件。因为只是用 UML 类图表示数据需求,并不意味着一定用面向对象编程实现代码。

3.8 分析之第五步——系统质量因素分析

系统质量因素属于第二类需求。GJB 438B 要求“若提出质量因素方面的需求,则本条应描述系统的这些需求,其中包括:功能性、可靠性、易用性、效率、维护性、可移植性以及其它属性等的定量需求。”

关于质量因素,最典型的问题是表达的经常是无效信息。这种信息无效性主要表现为两类:一类是照搬标准要求,例如照搬标准的易用性、可扩展性条款描述等,而不是按照标准要求进行质量因素分析的结果;另一类是直接将设计当成质量因素需求。

对质量因素进行分析,必须遵循万事皆可度量的原则,任何质量因素的分析结果都是可以量化的,避免出现笼统的、含糊的或放之四海而皆准的质量特性。

另外,质量因素分析不是针对某种设计提要求,质量因素分析仍属于“问题域”,是在提出问题,而不是给出解决方案。架构设计师需要针对质量因素给出恰当的解决方案。

3.8.1 质量因素分析方法

不同的领域,系统的质量因素千差万别。但是分析质量因素都需要从需求的三个层次分别入手。

首先,对来自出资方的业务级需求,这个层级的质量因素要充分考虑出资方研制系统的愿景。

其次,对于来自使用方的使用级需求,这个层级的质量因素要考虑运行期质量。

最后,对于来自开发方和保障方的开发级需求,这个层级的质量因素要考虑开发期质量。

遵循“万事皆可度量”的原则,所有分析得到的质量因素,无论是功能性、可靠性、易用性、效率、维护性还是可移植性等,都归结为以下三类。

(1) 第一类是可以定量度量的质量因素。

(2) 第二类是在架构设计初期,需要架构设计师转换为功能需求的质量因素。例如,访问安全性:软件应对三类用户角色分配相应的权限,禁止越权操作。该条质量因素可以被转换为一个系统用例:配置用户角色和权限。

(3) 第三类是在架构设计初期,需要架构设计师给出相应设计决策。例如:

① 可恢复性:系统宕机重启后,能够恢复至宕机前运行状态。其解决方案:关键数据自动保存。

② 容错性:软件应能够识别 CAN 总线数据的异常,防止异常数据造成软件崩溃。其解决方案为:对需要在 CAN 总线上交互的数据,在接口协议中设计 CRC 校验字段;软件读取 CAN 总线数据时进行 CRC 校验。

③ 易操作性:按设备名称检索历史记录时操作便捷,无须全名称录入。其解决方案:人机交互界面设计成带首字母筛选功能的下拉列表。

3.8.2 质量因素表达方法

质量因素的表述中还存在一类较为突出的问题：没有描述真实的质量因素需求，直接给出设计。

对系统规格说明中陈述的内容到底是需求还是设计，可以采用的一种评判方法，是用“不这样行吗？”^[1]的问句，就能够找出设计，然后问为什么，背后隐含的往往就是真实需求。

这个方法对质量因素的描述同样适用。例如：

(1) “系统采用冗余磁盘阵列”是质量因素需求还是设计？那么问需方“系统不采用冗余磁盘阵列行吗？”如果需方明确“不这样不行”，就是需求，否则就不是需求。显然，对这个问题，需方不可能做出肯定的回答，所以这不是需求。那么，接着问“为什么要这么做呢？”，原因是为了保证发生磁盘故障时降低对系统运行的影响，所以真实的需求是“磁盘故障平均发生间隔需大于 T 小时”。

(2) “系统在服务器端调用某种数据融合算法，把融合结果传回客户端”是质量因素需求还是设计？那么问需方“不这样行吗？”需方不会关注融合算法是什么，更不会关注融合算法在哪里实现，需方关注的是融合结果以及融合正确性。所以这不是需求。那么，接着问“为什么要这么做呢？”原因是使用者需要在 1s 内看到数据融合结果，而使用者不止一个，如果在每一个客户端上实现融合算法，客户端的硬件资源就需要高配。所以真实的需求是“使用者需要在 1s 内看到数据融合结果”，约束是“系统有 N 个使用者”。

(3) 同理，“界面设计如下图所示”这类表述也不是需求。

参考表 3-7，可以进一步理解质量因素的表述方法。

表 3-7 质量因素表述方法示例

表 述	责 任
对高铁自助售票系统,99%的顾客应该能在 5s 内,通过最少操作,完成自助出票	需求
什么样的交互界面能够满足最少操作的需求。	设计
什么样的系统构架能够满足 5s 内正确、安全地出票	
如何用界面构件实现这样的交互界面	编码

3.9 分析之第六步——设计和构造约束分析

设计和构造约束属于第三类需求。分为四类约束^[2]（四类约束 = 业务环境约束 + 使用环境约束 + 构建环境约束 + 技术环境约束），分别来自三类涉众，所以分别归属于需求的三个层次（业务级需求、用户级需求、开发级需求），见表 3-8。

表 3-8 四类约束与需求层次关系

需求层次	四类约束	
业务级需求	业务环境约束	技术环境约束
用户级需求	使用环境约束	
开发级需求	构建环境约束	

例如,某一线城市拟建设一座大型体育场馆,能够承接国际性体育赛事。

首先,考虑来自出资方的业务环境约束。出资方的建设愿景是提升城市形象。考虑这样的愿景带来的约束是场馆要兼具实用性和新颖性,能够成为城市中的一个地标性或景点式建筑。

其次,考虑来自使用方的使用环境约束。交通要便捷,无论是参赛运动员还是普通民众,都能够使用公共交通工具从各个方向便捷到达。由此带来的使用环境约束是具体地址选择。

之后,考虑来自建设方的构建环境约束。一旦地址选在城市中心区域,如何开展施工才能避免对周边居民生活产生影响。

最后,考虑技术环境因素。使用什么建筑技术能够在要求的时间内完成建设。

3.10 系统需求分析案例

本案例是以一个第三方软件测评机构为研究对象,该组织希望引进或改造一套管理系统,提升测试工作的管理效率。重点是针对测评过程涉及的大量技术记录进行自动化管理,这些技术记录管理的要求是记录必须有标识、相关记录之间必须可追踪。

下面按照系统需求分析方法,进行以下分析工作。

- (1) 确定待改进业务流程的组织——测评机构。
- (2) 发现组织的业务用例——软件测评。
- (3) 对业务用例的业务流程现状进行建模——软件测评的时序图(业务流程现状)。
- (4) 对改进后的业务流程进行建模——软件测评的时序图(业务流程改进)。
- (5) 确定系统用例——在时序图中,凡是指向新研/改进系统的消息均是系统职责。
- (6) 描述系统用例规约。

分析至此,得到了系统能力需求的第一类需求——功能需求,在此基础上,继续进行后续分析。

(7) 接口分析。这个比较简单,不再赘述。

(8) 质量属性分析。

(9) 设计约束分析。

3.10.1 对组织建模

对第三方软件测评机构而言,其组织的业务用例只有一个——软件测评,组织的业务用例图见图 3-12。

3.10.2 对组织业务流程现状建模

对软件测评业务用例使用 UML 序列图进行建模。

该组织开展软件测评业务主要涉及内部的业务工人(包括测试人员、配置管理人员、质量保证人员等)和业务实体(测试项目配置管理系统),使用 UML 的时序图对软件测评的业务流程建模,见图 3-13。

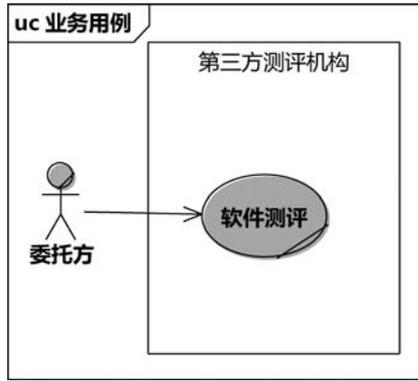


图 3-12 第三方测评机构的业务用例图

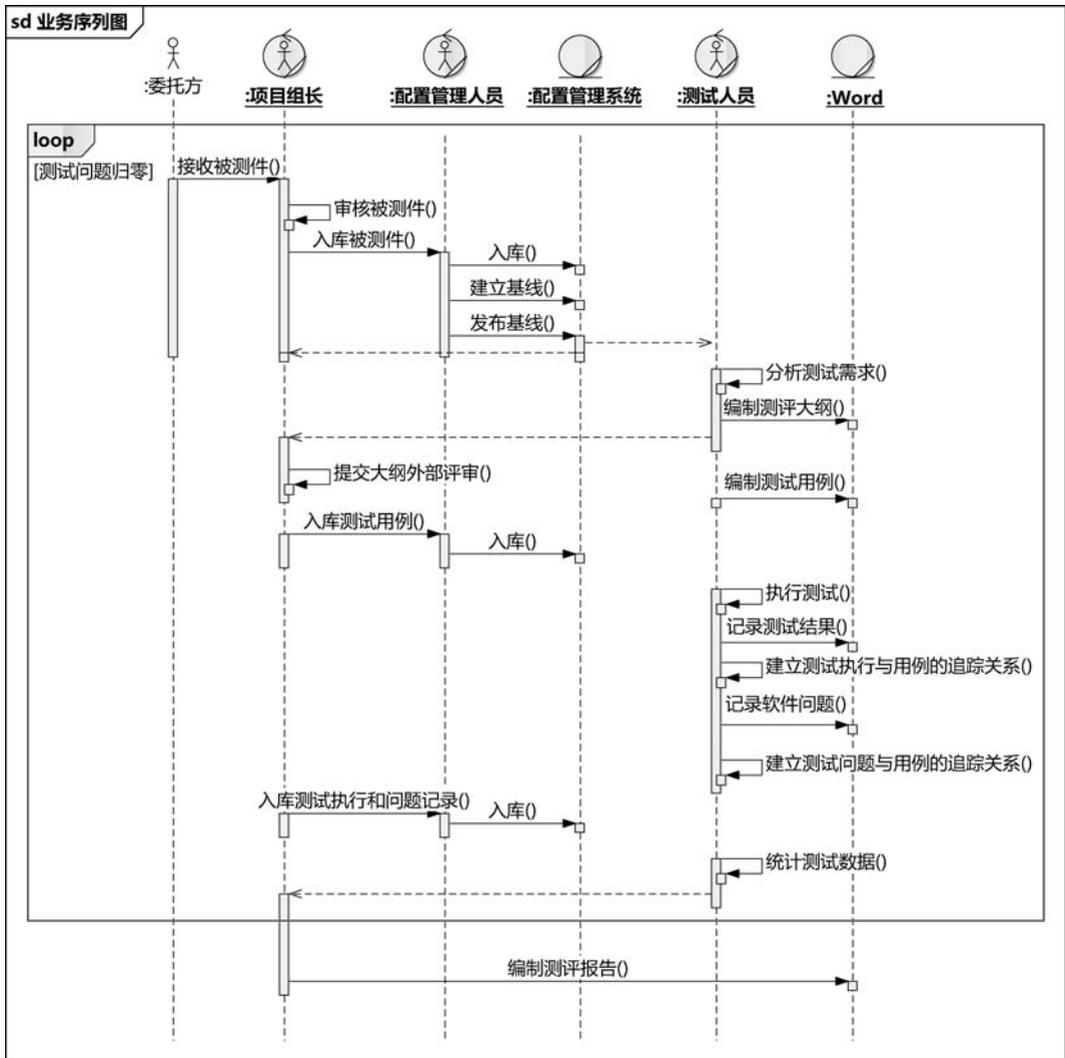


图 3-13 软件测评(业务用例)时序图——业务流程现状

软件测评业务流程现状如下。

- (1) 委托方向项目组长提交被测件,项目组长审核、提交配置管理人员入库。
- (2) 配置管理人员建立基线、发布。
- (3) 测试人员开展测试需求分析,编制测评大纲。
- (4) 测试组长提交外部评审。
- (5) 测试人员使用 Word 编写测试用例,为每个测试用例建立唯一标识。
- (6) 项目组长提交配管人员入库测试用例集合。
- (7) 测试人员执行测试用例,使用 Word 记录测试结果,为每个测试用例执行记录建立唯一标识。
- (8) 测试人员使用 Word 记录软件问题,为每个软件问题报告建立唯一标识,并追踪至测试用例。
- (9) 项目组长提交配管人员入库测试用例执行记录和软件问题报告集合。
- (10) 测试人员统计测试用例数量、测试用例执行数量、测试用例通过数量、软件问题数量。
- (11) 从第一步循环,开始下一轮测试,直至软件问题全部归零。
- (12) 项目组长编制测评报告。

从业务流程现状可以看出,测试人员承担了大量与测试技术无关的活动,包括为测试用例建立唯一标识、为测试用例执行记录建立唯一标识、为软件问题报告建立唯一标识、为软件问题报告和测试用例执行记录建立追踪关系、为测试用例执行记录和测试用例建立追踪关系;统计测试用例数、测试用例执行数、测试用例执行通过数、软件问题数等。

3.10.3 对组织业务流程改进建模

上述这些活动,不仅花费了测试人员大量的人力和时间,而且出错概率较高,为此,测评机构领导希望对现有的测试项目配置管理系统进行改进,使系统能够承担这些非技术性活动。

因此,希望改进的业务流程见图 3-14,将原测试项目配置管理系统升级为测试项目管理系统。改进后的业务流程如下。

- (1) 委托方向项目组长提交被测件,项目组长审核、提交配置管理人员入库。
- (2) 配置管理人员建立基线、发布。
- (3) 测试人员开展测试需求分析,使用 Word 编制测评大纲。
- (4) 测试组长提交外部评审。
- (5) 测试人员使用系统编写测试用例。
- (6) 项目组长使用系统将所有测试用例生成测试说明(Word 格式),系统为每个测试用例建立唯一标识。
- (7) 项目组长提交配置管理人员入库测试用例集合。
- (8) 测试人员执行测试用例,使用系统记录测试结果和软件问题。
- (9) 项目组长使用系统生成 Word 格式的测试用例执行记录、软件问题报告集合,系统为每个测试用例执行记录建立唯一标识、为每个软件问题报告建立唯一标识,并追踪至测试用例。

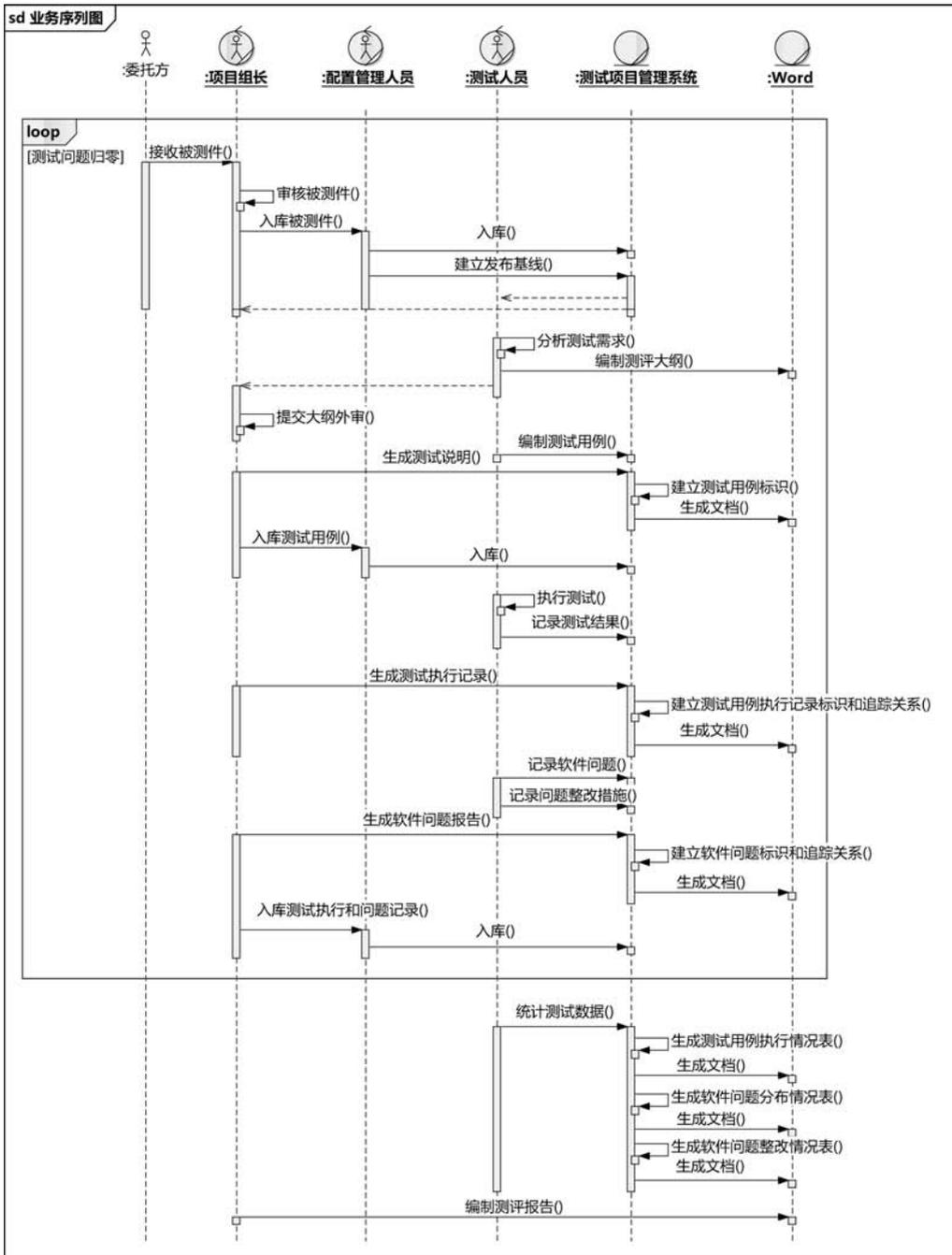


图 3-14 软件测评(业务用例)时序图——业务流程改进

- (10) 项目组长提交配置管理人员入库测试用例执行记录和软件问题报告。
- (11) 项目组长使用系统统计测试数据,能够生成以下 Word 格式表格: 测试用例执行情况表、软件问题分布情况表和软件问题整改情况等。
- (12) 从第一步循环,开始下一轮测试,直至软件问题全部归零。
- (13) 项目组长使用 Word 编制测评报告。

可见,系统改造升级前每个测试人员承担了很多手工编写标识的工作,不仅填写的记录多,且容易出错。改造升级系统之后,系统封装了业务逻辑,自动完成了上述工作,不仅保证了记录标识、追踪的正确性,而且大大减少了测试人员的工作量。

3.10.4 得到系统用例,确定系统规约

从图 3-14 中,得到测试项目管理系统的系统用例,使用 UML 用例建模,见图 3-15。具体的系统规约在此不再赘述。

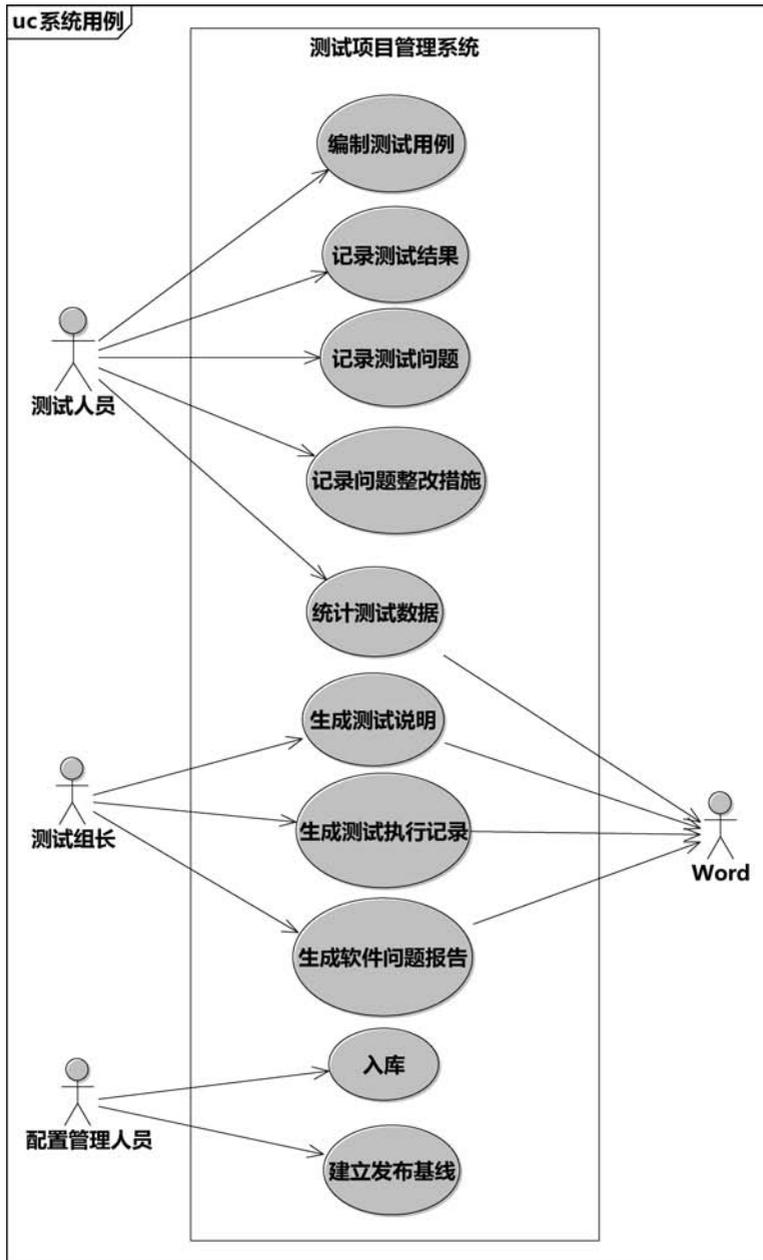


图 3-15 测试项目管理系统用例图

3.10.5 系统质量属性分析

下面首先对系统的质量属性从三个需求层次上展开分析,见表 3-9。

表 3-9 系统的质量属性分析

需求层次	质量属性类型	质量属性
业务级需求	商业质量	无
用户级需求	运行期质量	(1) 容量: 能够管理的一个项目的测试用例数量最多 5 万个 (2) 性能: 项目的测试用例达到容量时, 各类数据统计的时间均应小于 1s (3) 扩展性: 由于项目的测试轮次随实际情况而不同, 通常需要测试 2~3 轮次, 但是实际上不应该有上限限制, 系统应该对任意轮次的测试用例均能够管理 (4) 易用性: 由于测试人员存在一定流动性, 不希望系统在系统使用上花费公司太多培训精力, 最好不用培训, 就像目前测试人员使用 Word 编写测试用例一样, 无须对人员进行 Word 培训
开发级需求	开发期质量	适用性: 各类标识规则应该可配置, 避免变更标识规则时, 修改代码

3.10.6 系统设计约束分析

同样对系统的设计约束从三个需求层次上展开分析, 见表 3-10。

表 3-10 系统的设计约束分析

需求层次	设计约束类型	设计约束
业务级需求	业务环境约束	因为保密要求, 没有网络环境可以供系统使用
用户级需求	使用环境约束	对同一个项目, 测试现场比较分散, 测试人员不能集中开展测试
开发级需求	开发环境约束	需要与原配置管理系统无缝集成

3.11 系统规格说明文档模板解析

本节是对 GJB 438B 的系统/子系统规格说明模板的内容的详细解析, 展现了前文所描述的系统需求分析方法的分析结果如何落实体现在文档之中。

GJB 438B 的系统/子系统规格说明模板包括六个章节: 范围、引用文档、需求、合格性规定、需求可追踪性、注释。

本节内容除省略引用文档和注释外, 对其他章节内容逐一解析。

3.11.1 范围

范围的主要内容包括: 标识、系统概述、系统历史、项目的各相关方。

1. 标识

本条应描述本文档所适用系统的完整标识, 适用时, 包括其标识号、名称、缩略名、版本号 and 发布号。

- (1) 系统名称：按照合同中名称定义。
- (2) 系统标识：按照项目总体单位要求标识。
- (3) 系统版本：按照项目总体单位要求定义。
- (4) 系统简称：缩略名。

2. 系统概述

- (1) 系统所属的组织机构。
- (2) 该组织机构的职责。
- (3) 该组织机构中使用系统的业务工人角色。
- (4) 系统的主要用途。

3. 系统历史

概述系统开发、运行和维护历史。

系统分为两种情况：一是该类组织中以前没有此类系统，完全是新研制系统；二是该类组织中以前有此类系统，这次研制的系统是对原系统的改进。

对第一种情况，无系统历史信息。对第二种情况，应说明原系统的相关信息，包括研制总体单位、鉴定时间、运行期间发生的变更情况等。

4. 项目的各相关方

- (1) 需方：出资研制系统的机构。
- (2) 用户：最终使用系统的机构。
- (3) 使用总体：系统能力需求的论证机构。
- (4) 研制总体：系统的总承包方。
- (5) 开发方：系统的承研方和分承研方。
- (6) 保障机构：系统交付后负责保障维护的机构。

3.11.2 需求

需求是系统规格说明文档的主要内容。需求包括三类：功能需求、质量因素、设计约束。

在 GJB 438B 模板中的对应关系如下。

- (1) 模板的 3.2 系统能力需求是指功能需求。
- (2) 模板的 3.6 适应性需求、3.7 安全性需求、3.8 保密性需求和 3.11 系统质量因素、3.13 人员需求是质量因素。
- (3) 模板的 3.12 设计和构造的约束是指设计约束。
- (4) 模板的 3.1 要求的状态和方式是与系统使用要求(用途)密切相关的要求。
- (5) 模板的 3.3 系统外部接口需求、3.4 系统内部接口需求和 3.5 系统内部数据需求是支撑功能需求的接口需求和数据需求。
- (6) 模板的 3.9 系统环境需求和 3.10 计算机资源需求是指系统运行所依赖的计算机软件 and 硬件资源需求。
- (7) 模板的 3.14 培训需求、3.15 保障需求、3.16 其他需求、3.17 包装需求等主要是指支持系统交付的需求。

1. 要求的状态和方式

GJB 438B 要求“如果要求系统在多种状态或方式下运行,并且不同的状态或方式具有不同的需求,则应标识和定义每一种状态和方式”。注意,此要求的重点是两个条件同时成立,即不仅有多种运行状态,而且每种状态下的系统需求不同,此时,这条必须说明以下内容。

(1) 系统有多种运行方式和状态时,可先用表 3-11 说明方式和状态的关系。

表 3-11 要求的状态和方式(示例)

	方式 1	方式 2	方式 3
状态 1	√		
状态 2		√	
状态 3		√	√

(2) 用状态图或文字说明这些状态/方式之间如何转换。

(3) 系统有多种运行方式或状态时,需按表 3-12 说明系统的能力需求在哪个状态/方式下有效。

表 3-12 要求的状态和方式与系统能力关系(示例)

	状态 1	状态 2	状态 3
用例 1	√		√
用例 2		√	

2. 系统能力需求

系统能力需求的主要表述内容包括:一个系统用例图,以及每个系统用例的规约表。

(1) 如果本系统规模较大,且各部分之间有明确的职责区域,则这些部分可以直接划分为子系统,在此处用 UML 构件图说明这些子系统之间的关系,如图 3-16 所示。切记,此处不能够直接给出系统内部的详细部件组成,部件及其组成关系是系统设计范畴内的内容。



图 3-16 系统的子系统构成(示例)

(2) 用如表 3-13 所示的表格说明子系统用途(职责)。之后,后续章节按照子系统组织,也可以对各个子系统出具单独的子系统规格说明文档。

表 3-13 系统构成表(示例)

子系统名称	子系统标识	子系统用途	技术状态
			新研/改进/选用

(3) 如果本系统无须划分为多个子系统,或暂时无法明确子系统边界,需要在系统设计阶段确定的,直接画出系统的 UML 用例图,并用列表概述系统用例,如图 3-17 和表 3-14 所示。

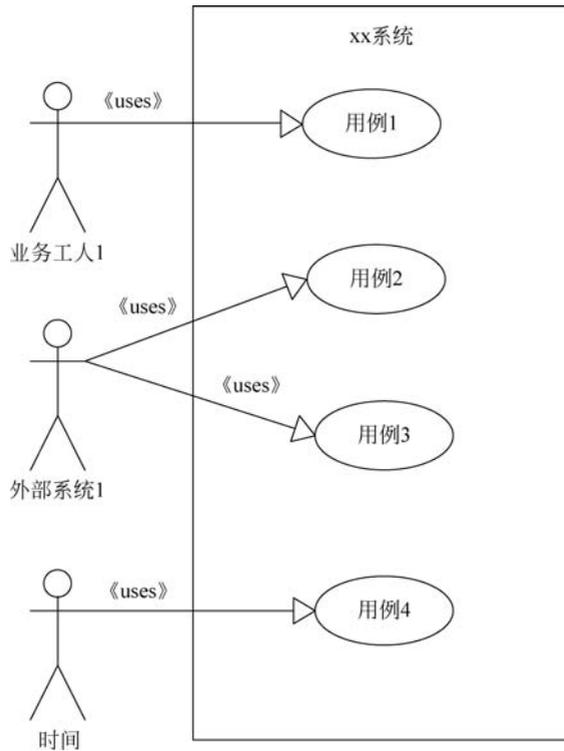


图 3-17 系统用例图(示例)

表 3-14 系统用例列表(示例)

序号	用例名称	标识	功能描述	技术状态
1	用例 1			新研/改进/复用
2	用例 2			

(4) 按照模板要求,以每个系统用例为小章节分条说明系统能力需求。每个系统用例使用如表 3-15 所示进行描述。

表 3-15 系统用例规约表(示例)

用例名称	动宾结构命名	项目唯一标识符	
研制要求章节			
简要描述	用例目标简述: 概述用例的价值 //注意: 此处的描述不要超出用例的职责范围		
参与者	主执行者: 必须有主执行者 辅助执行者: 视具体情况, 可以没有辅助执行者。如果系统自身能够完成该用例职责, 不需要系统外部的实体对象帮忙, 就没有辅助执行者		
前置条件	该用例开始前, 系统需要满足的约束, 且一定是系统能够检测到的 注意: 此处是指这个特定的用例开始执行前, 系统必须满足的条件或状态; 如果不满足这个前置条件, 这个用例就不能开始执行		

续表

	步骤	描述
主流程(代表用例核心价值的路径)	1	第一步的主语一定是主执行者
	2	(1) 一定聚焦于系统与外部的交互过程 (2) 每个步骤不要没有主语,且主语只能是主执行者或系统 (3) 使用主动语句,突出主语所承担的责任 (4) 使用系统所属组织的领域(核心域)的词语 (5) 不要描述交互过程中设计的细节 (6) 不要描述系统不能负责的事情
扩展流程	1a	对应流程中某个步骤中,系统要处理的意外和分支
	1a-1	针对 1a 所描述的意外,系统的处理流程
子流程	对应主流程中多次重复的一组步骤集合	
后置条件	该用例结束后,系统需要满足的约束,且一定是系统能够检测到的	
规则与约束	(1) 质量属性(性能等) (2) 设计约束: 业务规则、数据约束、技术约束	

3. 系统外部接口需求

(1) 画出系统或子系统与系统外部的其他硬件和软件之间的接口,如图 3-18 所示。



图 3-18 系统外部接口图(示例)

(2) 用表格说明每个接口的标识、类型、用途等,如表 3-16 所示。注意: 如果与外部系统存在多个物理接口,如网络、CAN 接口,需要在图和表中按不同接口分别说明。

表 3-16 系统外部接口需求表(示例)

序号	接口名称	标识	接口类型	接口用途	外部接口实体	外部实体状态
			网络、串口、CAN 总线、IO 等		外部系统 1	新研、沿用、改进

(3) 对识别出来的每个接口分别按条描述,如表 3-17 所示。

表 3-17 外部接口需求表(示例)

接口名称	某接口		接口标识	JK-OU-001
接口实体	接口两端的软件或硬件		接口类型	网络等
接口用途	简述接口两端的软件或硬件使用该接口的用途			
接口数据	(1) 说明该接口上传数据的种类 (2) 说明该接口上传的数据的编码格式: 如果每类数据使用相同的编码格式,用其他的表说明该格式; 否则,逐一说明每类数据使用的编码格式; 如果编码格式为标准格式,说明即可 (3) 说明每类数据的具体定义。如果数据较多,可以在附录中或使用 GJB 438B 的《接口设计说明》模板详细说明			
接口通信特征	通信链路	如网络的带宽、串口速率等		
	数据传输	非/周期性、单向或双向传输		
接口协议特征	说明接口是否有同步机制,详细说明同步机制如何设计的接口上的每类数据的传输层协议等			

注意：此时尚处于系统需求分析阶段，接口重点关注数据内容、数据取值的物理意义范围，不关心在计算机编程层面定义的数据格式，如数据类型（浮点、整型、字符串等）、字节组织形式等；这些内容属于对接口需求的具体解决方案，留待系统设计时确定。

4. 系统内部接口需求

系统内部接口需求，主要是指已经明确的子系统之间的接口。注意，不应该识别出系统的部件（软件和硬件）之间的接口，因为在系统需求分析阶段，系统设计还未出结果，系统的部件还不明确。

如果本系统无多个子系统，或暂时无法明确子系统边界，需要在系统设计阶段确定的，此处可留待设计时描述。

如果本系统已经明确划分为多个子系统，则参照系统外部接口的分析方法，标识子系统之间的接口。

(1) 画出子系统之间的接口，如图 3-19 所示。

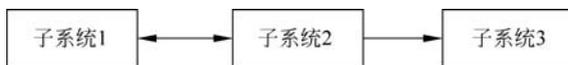


图 3-19 系统内部接口图(示例)

(2) 用表格说明每个接口的标识、类型、用途等。注意：如果两个子系统之间存在多个物理接口，如网络、CAN 口，需要在图和表中按不同接口分别说明。

(3) 分条目分别使用表 3-18 描述每个内部接口的需求。注意，内部接口的接口实体应该是子系统。

表 3-18 内部接口需求表(示例)

接口名称	某接口	接口标识	JK-OU-001
接口实体	接口两端的其他子系统	接口类型	网络等
接口用途	简述接口两端的子系统使用该接口的用途		
接口数据	(1) 说明该接口上传输数据的种类 (2) 说明该接口上传输的数据的编码格式：如果每类数据使用相同的编码格式，用其他的表说明该格式；否则，逐一说明每类数据使用的编码格式；如果编码格式为标准格式，说明即可 (3) 说明每类数据的具体定义。如果数据较多，可以在附录中或使用 GJB 438B 的《接口设计说明》模板详细说明		
接口通信特征	通信链路	如网络的带宽、串口速率等	
	数据传输	非/周期性、单向或双向传输	
接口协议特征	说明接口是否有同步机制，详细说明同步机制如何设计的接口上的每类数据的传输层协议等		

5. 系统内部数据需求

识别系统内部需求的目的是为数据库表/文件的设计提供需求来源。识别方法是从各个系统用例中找出系统需要处理的业务实体。建模方法是使用 UML 类图对实体类建模。注意，内部数据需求不是输入、输出数据。

(1) 建立系统的实体类图。

(2) 使用表 3-19 说明识别出的所有实体类。

表 3-19 系统的实体类列表(示例)

序 号	类 名 称	标 识	相关的系统用例
			是指该类在哪些系统用例中出现

(3) 对每个实体类分条使用表 3-20 进行说明。注意：此处关注类的属性,不关注类的操作,类的操作是系统设计阶段的内容。

表 3-20 某某类(示例)

序号	属性名称	取值范围	精度要求	组成格式	是否为公共属性
		注意：是有物理意义的取值范围,如电压范围、高度范围等		使用数据字典的词条法表示：=、+、[/]、()、* ... *、{ } ^{n(重复次数)} 例如：编号由 2 位字母、6 位数字和 1 个可选 x 组成。表示如下： 编号 = {[A...Z/a...z]} ² + {0...9} ⁶ + (x)	系统的公共数据的来源

6. 适应性需求

适应性需求是指系统在不同用户现场安装时需要修改的安装数据,如地理数据配置文件、菜单配置文件、通信端口参数等。可以用表 3-21 表示。

表 3-21 适应性需求表(示例)

序 号	名 称	类 型	内 容	用 途
		文件、数据库表、注册表		

7. 安全性需求

安全性需求是指系统可能涉及的两类安全风险：一类是可能导致系统毁坏或人员伤亡的安全风险；另一类是指可能被恶意入侵的风险。

注意以下三点常犯错误：一是不要抛开安全性需求的基本原则,为了写安全性需求而硬写,随意扩大安全性需求内涵,分析出一些并不适用的安全性需求；二是没有针对软件进行安全性分析,复制一些放之四海而皆准的要求；三是混淆了需求和解决方案,直接写出解决方案,而忽视了真正的需求。

8. 保密性需求

GJB 438B 原文要求“若有保密性需求,本条应指明维持保密性的系统需求,包括：系统运行的保密性环境、所提供的保密性的类型和级别、系统必须经受的保密性风险、减少此类风险所需的安全措施、必须遵循的保密性政策、系统必须具备的保密性责任、保密性认证/认可必须满足的准则等。”

此条是指系统交付后,系统在使用环境中的保密性要求。如与外部系统交互时,数据需加密交互,确保数据完整性、不可抵赖性等；系统保存的关键数据是否需要加密存储；对关键数据的销毁的要求等。注意,不要描述开发单位的保密管理要求,以及开发环境的保密性要求等。

9. 系统运行环境需求

GJB 438B 原文要求“若有系统环境需求,本条应指明与系统运行必需的环境有关的需求。对软件系统而言,运行环境包括支持系统运行的计算机硬件和操作系统。对硬软件系统而言,运行环境包括系统在运输、存储和操作过程中必须经受的环境条件,如自然环境条件(风、雨、温度、地理位置)、诱发环境(运动、撞击、噪声、电磁辐射)和对抗环境(爆炸、辐射)。”

对纯软件系统而言,此时应按要求说明支持系统运行的计算机硬件和操作系统。

对软件和硬件结合的系统而言,此时,应按要求说明对使用环境的需求;由于此时还没有确定软件能力需求,所以对软件的运行环境可以留待系统设计时明确。

系统运行环境需求表示例如表 3-22 所示。

表 3-22 系统运行环境需求表(示例)

序号	硬件名称	类 型	软件名称	类 型
		处理器、存储器、输入/输出设备、辅助存储器、通信/网络设备		操作系统、地理信息系统、数据库系统、Office 办公软件、通信软件、框架软件、其他第三方软件等

10. 计算机资源需求

1) 计算机硬件需求

GJB 438B 原文要求“本条应描述系统使用的或引入到系统中的计算机硬件的需求,包括:各类设备的数量;处理机、存储器、输入/输出设备、辅助存储器、通信/网络设备及所需其他设备的类型、大小、容量和其他所需的特征。”

此条对纯软件系统直接有效,应该按要求对系统环境需求中的计算机硬件需求进行详细说明,如表 3-23 所示。

表 3-23 计算机硬件需求表(示例)

序号	硬件名称	类 型	资源配置说明	数 量	来 源
		处理机、存储器、输入/输出设备、辅助存储器、通信/网络设备			

2) 计算机硬件资源使用需求

GJB 438B 原文要求“本条应描述本系统的计算机硬件资源使用需求(若有),例如,最大允许利用的处理机能力、内存容量、输入/输出设备的能力、辅助存储设备容量和通信/网络设备的能力。这些需求(例如陈述为每一个计算机硬件资源能力的百分比)应包括测量资源使用时所处的条件(若有)。”

此条对纯软件系统直接有效,应该按要求对计算机硬件需求中的各个硬件资源使用的需求进行详细说明,如表 3-24 所示。

表 3-24 计算机硬件资源使用需求表(示例)

序号	硬件名称	类 型	使用要求	备 注
		处理机、存储器、输入/输出设备、辅助存储器、通信/网络设备	软件使用内存余量等	

3) 计算机软件需求

GJB 438B 原文要求“本条应描述本系统必须使用或引入系统的计算机软件的需求(若有)。例子包括：操作系统、数据库管理系统、通信/网络软件、实用软件、输入和设备仿真软件、测试软件和制造软件。要列出每一个这样的软件项的正确名称、版本和参考文档。”

此条对纯软件系统直接有效,应该按要求对系统环境需求中的各个软件需求进行详细说明,如表 3-25 所示。对软件和硬件结合系统,如果此时还不能够确定系统对其他软件的需求,可以留待系统设计时明确。

表 3-25 计算机软件需求表(示例)

序号	软件名称	类 型	版 本	来 源	备 注
		操作系统、地理信息系统、数据库系统、通信/网络软件、实用软件、输入和设备仿真软件		自备、自研、货架、第三方(某公司)提供	

4) 计算机通信需求

GJB 438B 原文要求“本条应描述本系统必须使用的计算机通信方面的需求(若有)。例子包括：要连接的地理位置；配置和网络拓扑；传输技术；数据传送速率；网关；要求的系统使用时间；被传送/接收的数据的类型和容量；传送/接收/响应的的时间限制；数据量的峰值；以及诊断特性。”

这条要求应该是指系统在使用环境中,与外部系统进行通信的需求,而不是系统内部的通信需求。例如,系统通过什么配置的网络交换机与外部系统通信,分配给系统使用的带宽和使用时间是否有要求；系统通过什么通信能力的电台与外部系统通信,最大话音传输能力等。

11. 系统质量因素

GJB 438B 原文要求“若提出质量因素方面的需求,本条应描述系统的这些需求。包括：功能性、可靠性、易用性、效率、维护性、可移植性和其他属性的定量要求。”

提示：系统质量属性是指用户级需求层面的质量属性。用户级层面的质量属性包括 6 类 21 项质量特性,这些质量属性可以通过分析业务级需求中的功能需求、质量需求、约束,以及用户级需求中的约束得到。

示例如表 3-26 所示。

表 3-26 系统质量因素分析表(示例)

需求层次	功 能 需 求	质 量 属 性	约 束
业务级需求	从业务愿景中分析质量属性。如“适应业务快速变化”分析得到“易改变性”质量属性		从业务环境约束中分析质量属性。如“与原有办公系统集成”分析得到“互操作性”质量属性
用户级需求		从业务级需求和用户级需求的“约束”中分析得到以下质量属性。 (1) 易改变性 (2) 互操作性 (3) 易用性 (4) 容错性	从使用环境约束中分析质量属性。如“操作人员是非计算机专业出身,计算机水平普遍不高”分析得到“易用性”和“容错性”质量属性

12. 设计和构造约束

提示：设计约束包括四类：业务环境约束+使用环境约束+构建环境约束+技术环境约束；这四类约束分布在不同层面的需求。

业务环境约束。属于业务级需求，来自出资方的约束，例如上线时间、预算、集成、业务规则、行业法律法规（禁止使用解释性编程语言）等；由出资方指定技术选型（某平台、必须使用某数据库系统、必须遵循某数据标准、应与原某系统集成、应与某系统互连互通等）。

使用环境约束。属于用户级需求，来自使用方的约束，如使用者的专业能力、何种人群、分布式使用、使用环境有电磁干扰、车船移动等因素。

构建环境约束。属于开发级需求，来自开发和维护人员的约束，如开发人员的技术水平、业务知识、管理水平等。

技术环境约束。来自业界当前技术环境约束，如成熟算法、技术平台、中间件、编程语言成熟度等。

这四类约束可以分为以下三种情况。

(1) 直接制约设计决策的约束。如系统运行于 UNIX 平台上。

(2) 转换为功能需求的约束。如应严格执行总部统一规定的商品折扣率。分析后可转换为功能需求：调整商品折扣率。

(3) 转换为质量属性的约束。如操作人员计算机水平普遍不高。分析后转换为系统应具有高易用性（如完成一个业务操作平均输入数据次数最多 N 次）和容错性（如系统对输入数据在人机交互界面进行有效限制，避免操作人员输入非法数据。）

13. 人员需求

GJB 438B 原文要求“若有人员相关要求，则本条应描述与使用或支持系统的人员有关的需求”。

注意：本条不应该描述与开发人员相关的需求，也不是单纯描述系统使用人员的数量、技能需求。

本条的目的是通过对系统使用或支持人员的数量、技能等信息进行分析，得到对系统质量因素方面的需求。参考人素工程的定义，“人素工程是一门多学科的交叉学科，研究的核心问题是不同的作业中人、机器及环境三者之间的协调……研究的目的是通过各学科知识的应用，来指导工作器具、工作方式和工作环境的设计和改造，使得作业在效率、安全、健康、舒适等几个方面的特性得以提高。”

14. 培训需求

对系统而言，对系统使用人员进行培训是有必要的。所以，本条应该描述与培训相关的需求。例如，需要培训的系统操作人员（角色）、数量、培训使用的教材（包括使用手册、用户手册、电子交互手册，以及其他教材）等。

注意：本条不是对系统使用方提出要求，而是站在需方的角度，对系统研制方提出培训需求，要求研制方对系统使用方进行培训，要求研制方提供合适有效的培训资料。

15. 保障需求

GJB 438B 原文要求“若有综合保障相关需求，则本条应描述有关综合保障方面的系统需求，其中包括系统维护、软件保障、系统运输方式、对现有设施的影响和对现有设备的影响”。

本条的目的是通过对系统交付后,如何使得保障机构能够保障系统正常运转进行分析,得到与保障工作相关的系统质量因素方面的需求。

注意:本条不是对系统保障机构提要求,而是站在需方的角度,对系统研制方提出要求,要求研制方基于系统的维护保障工作的顺利实施,分析系统需要承担的职责。

16. 其他需求

GJB 438B 原文要求“若有其他需求,则本条应描述在以上各条中没有涉及的其他系统需求,……”。

这条的目的是如果需方还有在合同中,以及上述需求中未覆盖到的其他系统需求,包括研制方应提供的系统文档需求,可以在本条进行说明。

17. 包装需求

GJB 438B 原文要求“若有包装需求,则本条应描述需交付的系统及其部件在包装、加标签和处理方面的需求,适当时可引用适用的军用规范和标准”。

这条的目的很明确,站在需方的角度,对系统交付时的包装提出要求。一是系统及其部件的包装要求;二是系统及其部件的外加标签要求。

18. 需求的优先顺序和关键程度

GJB 438B 原文要求“本条应描述本规格说明中各需求的优先次序、关键性或所赋予的指示其相对重要性的权重”。

这条包含两方面的要求,一是需求的优先次序,优先次序是研制方优先实现系统需求的依据;二是需求的关键性,关键性是研制方需要特别关注是否需要特殊处理(如需要高端专业人员参与研制、系统设计时需要给出专门的设计决策等)的依据。

例如,优先顺序定义为 1 和 2。1 是需要优先实现的基本需求;2 是增强需求。

关键性通常是从系统安全性和保密性角度考虑的,建议由高至低分为三个等级(A、B、C)。

A 是关键程度最高的需求,是指如果该类需求失效,直接影响系统所属组织无法完成核心业务或危害到系统和人员的安全。

B 是关键程度第二等级的需求,是指如果该类需求失效,影响系统的主要性能实现。

C 是关键程度最低等级的需求,是指除了 A 和 B 等级外的需求。

需求优先顺序和关键程度列表例如表 3-27 所示。

表 3-27 需求优先顺序和关键程度列表

序 号	需求名称	需求标识	优先顺序	关键程度

3.11.3 合格性规定

GJB 438B 原文要求“本条应定义一组合格性方法,并为第 3 章中的每个需求指定为确需求得到满足所应使用的方法”。

合格性规定是系统合格性测试的依据。按照 GJB 2786A 的要求,系统合格性测试应具有独立性(即必须由独立于系统研制团队的人员承担系统合格性测试)。本条的目的是站在

需方的角度,为每条系统需求提出验证其合格性的方法。

GJB 438B 提出的合格性方法如下。

(1) 演示:依靠可见的功能操作,直接运行系统或系统的一部分,而不需要使用仪器、专用测试设备或进行事后分析。

(2) 测试:使用仪器或其他专用测试设备运行系统或系统的一部分,以便采集数据供事后分析使用。

(3) 分析:处理从其他合格性方法获得的累积数据。

(4) 审查:对系统部件、文档等进行目视检查。

(5) 特殊的合格性方法:任何针对系统的特殊合格性方法,如专用工具、技术、规程、设施、验收限制、飞行模型或飞行航迹等。

合格性规定示例如表 3-28 所示。

表 3-28 合格性规定

序号	需求名称	需求标识	合格性方法
1			演示
2			使用某测试设备进行测试
3			分析某测试结果数据
4			审查某文档

3.11.4 需求可追踪性

使用正向追踪表,说明研制要求的每项要求被分解给了哪些系统需求(系统规格说明中的三类需求)。

使用逆向追踪表,说明每个系统需求(系统规格说明中的三类需求)来自哪些研制要求。

思考题

题目 1 对附录 1 的数据采集系统案例,本书没有按照先找组织、发现组织的业务用例、从业务用例序列图发现系统用例的方法确定系统用例,请问如果按照上述方法,该系统的用例能够从业务流程中得到吗? 如果不可以,还有其他途径得到吗?

题目 2 一个交警大队,其组织的业务用例通常包括什么?

题目 3 组织的业务用例和系统用例有何关系?