



西安交通大学  
XI'AN JIAOTONG UNIVERSITY

## 分析总报告

需求管理系统分析

课程名称: 软件系统分析与设计

姓名: 凌晨

学院: 软件学院

专业: 软件工程

学号: 2214414320

2023 年 9 月 27 日

## 一、 目的和要求

请结合教材相关内容以及网络资料与课件，在对需求属性特征进行定义的基础上，对网络中任意提供的一个需求管理系统进行分析，或者基于个人对需求工程的理解，请尝试对整个需求管理系统中的实体以及属性进行建模，并对整个系统的开发与实现进行设计。(注：可以采用 ER 图，也可以采用类图或其它模型来表示。

## 二、 需求工程流程

为了探究需求管理系统，我们现需要深入了解需求工程的流程。简单地进行分解，可以将需求工程流程分为以下 5 个阶段：

### 1. 需求导出

需求工程的第一步为需求导出，也称需求获取，是需求工程最重要的基石。

需求的导出具有多种方式，是无数项目经理在不断地与甲方沟通探索中发明出来的。一般而言，有用户面谈、需求专题讨论会、问卷调查、现场考察等实际方法。但是，需要注意，即使是科学有效的方法获取需求，因为甲方并不都是了解技术或者并不完全理解自己想要什么功能需求，因此得到的需求通常是模糊的。所以，当获取得到调查结果后，要及时保留完整的原始信息，比如过程信息，包括但不限于时间地点方式等，再比如调查结果文档。得到的原始信息需要保存在系统中。

### 2. 需求挖掘

前文已经说过，通过甲方得到的需求往往是不完整和难以实现的。因此，当获取到需求以后，自然需要进行需求的挖掘。需求的挖掘主要分为两个方面展开，需求分析和需求定义，注意这两个方面不是完全割裂的。

需求定义为利用我们得到的调查文档，提取重要信息，将原本模糊的需求细致化，对需求类型，优先级，重要性，可行性，风险级别，需求详情进行确认。

需求分析为在定义完的基础上思考甲方深层次需求，有效地更改个别需求细节，确定甲方需求痛点，保证需求既不画蛇添足，也不缺斤少两。

### 3. 需求确认

当需求挖掘完成以后，需要开发方和客户方共同对《产品需求规格说明书》进行评审，达成共识后作出承诺。当需求作废时，要重新进行需求分析。此时我们的系统需要记录这一过程，信息包括项目名称，需求名称，开发方代表姓名，客户方代表姓名，时间，地点，确认状态等。

### 4. 需求开发及变更

确定需求完毕，就要进行需求开发了，开发过程不必赘述。

而需求变更是需求工程的一大难点。在现实中，总会存在原先需求文档存在错误或不足以及原先的需求文档无法满足当前市场需求，导致的需要变更需求。因此，系统也需要记录这一信息。其中信息包括变更类型，提交日期，更新日期，描述，实现优先级，修改者，建议者，建议者设置的优先级，实现版本，项目名称，变更详情，状态，验证者等。同时，在需求变更的时候也不能忘记了前两步。

## 5. 需求跟踪

需求跟踪是需求管理的一项重要内容。指跟踪一个需求使用期限的全过程，需求跟踪包括编制每个需求同系统元素之间的联系文档，这些元素包括：其他类型的需求，体系结构，其他设计部件，源代码模块，测试，帮助文章等。

其中实现需求分析的方式有多种：需求跟踪的实现方法：

- 需求跟踪矩阵 (RTM)：主要验证需求是否得到了实现的有效工具，借助该矩阵，可以跟踪每个需求的状态的状态（最常用的方法）
- 实体联系模型：是使用实体联系模型来描述需求的跟踪联系
- 交叉引用：主要被用来在文档之间建立跟踪联系。如系统需求规格说明文档、软件需求规格说明文档等。

## 三、 再论需求工程流程

我们刚刚简要地进行了需求工程流程的讲解，已经有了初步的概念与对需求管理系统的设想。但是要细化出一张张二维表建立出 ER 图还需要对每个阶段进行更为详细的分析，才能设计出需求管理系统。

### 1. 需求导出

#### (1) 需求来源信息

现实错综复杂，为了得到准确无误的需求也需要采用不同的有效方法。同时，调查需求的会留下多种关键信息，这些信息对需求优先级等方面有着重要影响。设计表格如下：

列名	含义
rs_id(PK)	原始需求标识符
method_id(FK)	调查方法识别符
rs_time	需求获取时间
rs_place	需求获取地址
rs_name	需求简称
rs_file_link	需求原始描述文档地址
s_id(FK)	需求调查人员标识符
u_id(FK)	需求提供人员标识符

表 1: 需求来源信息表 requirement\_source

#### (2) 员工、甲方

由于需求还未完全给出，暂且不能给出设计表，但这里可以提前定义需求调查人员（员工）需求提供者（甲方）的设计表格：

列名	含义
s_id(PK)	员工标识符
s_name	员工名字
s_phone	员工联系电话
d_id(FK)	员工所属部门
s_work	员工职责
s_mail	员工邮箱
s_age	员工年龄
s_gender	员工性别
...	...

表 2: 员工表 staff

注意，还有... 是指需要根据实际公司需求增加员工的其他属性，这里不再一一列举。

列名	含义
u_id(PK)	甲方标识符
u_name	甲方名字
u_phone	甲方联系电话
u_company	甲方公司
u_mail	甲方邮件
u_age	甲方年龄
u_gender	甲方性别
...	...

表 3: 甲方表 user

注意，还有... 是指需要根据实际公司需求增加甲方的其他属性，这里不再一一列举。

### (3) 调查方法

列名	含义
method_id(PK)	调查方法识别符
m_name	调查方法名字
m_description	调查方法说明
m_priority	调查方法优先级

表 4: 调查方法表 method

### (4) 部门信息

在员工表中，存在部门的信息，在此可以直接给出部门设计表格：

列名	含义
d_id(PK)	部门标识符
d_name	部门名字
d_phone	部门联系电话
d_company	部门职责
...	...

表 5: 部门表 department

注意，还有... 是指需要根据实际公司需求增加部门的其他属性，这里不再一一列举。

## 2. 需求挖掘

### (1) 需求定义

需要将通过各种方式获得到的原始需求转化为符合软件工程的需求，这需要对需求进行定义，得到了以下几个维度：

列名	含义
r_id(PK)	需求标识符
r_version	需求版本
r_name	需求名称
r_description	需求简述
r_file_link	需求详细描述文档地址
r_priority_id(FK)	需求优先级
r_status_id(FK)	需求状态
r_feasibility_id(FK)	需求可行性
r_kind_id(FK)	需求类型

表 6: 需求定义表 requirement

注意：在需求定义表中有多个属性描述需求，为了避免数据冗余和开发需要，需要对一些属性单独建立表：

列名	含义
r_priority_id(PK)	需求优先级
r_priority_name	名称
r_priority_description	简述

表 7: 需求优先级表 r\_priority

列名	含义
r_status_id(PK)	需求状态
r_status_name	名称
r_status_description	简述

表 8: 需求状态表 r\_status

列名	含义
r_kind_id(PK)	需求类型
r_kind_name	名称
r_kind_description	简述

表 9: 需求类型表 r\_kind

列名	含义
r_feasibility_id(PK)	需求可行性
r_feasibility_name	名称
r_feasibility_description	简述

表 10: 需求可行性表 r\_feasibility

注意：在定义的时候，总归是有定义人员的，而且不同版本有着不同的定义人员，确定定义人员可以实施严格的追查制度。因此需要一个表，可以简称为责任管理表格如下：

列名	含义
r_id(PK+FK)	需求标识符
s_id(PK+FK)	需求定义人员
...	...

表 11: 责任管理表 duty

... 为责任管理定义表的其他属性，需要根据具体的公司制度制定，不再一一列举出来。

## (2) 需求分析之角色

在软件工程中，实现需求不是为某一特定用户实现的，而是对一类用户实现的，这一类用户可以合并为角色，下面为角色表：

列名	含义
ro_id(PK)	角色标识符
ro_name	角色名称
rp_id(FK)	角色特权识别符
ro_description	角色简述
...	...

表 12: 角色表 role

注意，还有... 是指需要根据实际需求增加角色的其他属性，这里不再一一列举。

为了区分角色特权，还需要建立角色特权表：

列名	含义
rp_id(PK)	角色特权识别符
rp_name	特权名称
rp_description	特权简述
rp_file_link	特权详细描述文件链接

表 13: 角色特权表 role\_privilege

### (3) 需求分析之需求关系

需求之间也有各种各样的关系，比如有的需求是其他需求的子需求或者父需求，有的需求要满足前继或者后继关系，因此需要一个表专门记录需求关系，以便开发人员进行查阅编码。

列名	含义
r_id_first(PK+FK)	第一需求识别符
r_id_second(PK+FK)	第二需求识别符
rr_id(PK+FK)	需求关系识别符

表 14: 需求关系表 requirement\_relation

还需要定义需求关系定义表

列名	含义
rr_id(PK)	需求关系识别符
rr_name	需求关系简称
rr_description	需求关系简述
rr_difficulty	需求关系实现难度

表 15: 需求关系定义表 requirement\_relation\_define

## 3. 需求确定

在需求导出和需求挖掘完成后，需要进行需求确定，下面为需求确定表：

列名	含义
v_id(PK)	需求确定识别符
v_name	需求确定名称
v_time	需求确定时间
v_place	需求确定地点
v_file	需求确定文件链接
s_id(FK)	员工识别符
u_id(FK)	甲方识别符
...	...

表 16: 需求确定表 validation

注意，还有... 是指需要根据实际合同需求增加其他属性，这里不再一一列举。  
还需要建立需求和需求确立的关系表：

列名	含义
v_id(PK+FK)	需求确定识别符
r_id(PK+FK)	需求识别符

表 17: 需求-需求确定表 r\_v

#### 4. 需求开发及变更

##### (1) 需求开发

需求开发往往需要复杂的文档确立如何开发需求，为了简化起见，设置了以下属性：

列名	含义
ri_id(PK)	需求开发识别符
ri_implement	需求开发接口
ri_file	需求开发文件链接
ri_status(FK)	需求开发状态
start_time	需求开发开始时间
last_time	需求开发预计时间
ri_priority(FK)	需求开发优先级

表 18: 需求开发表 requirement\_implement

需要对需求开发状态单独建立表格：

列名	含义
ri_status(PK)	需求开发状态
ri_status_name	名称
ri_status_description	简述

表 19: 需求开发状态表 ri\_status



需要对需求开发优先级单独建立表格：

列名	含义
ri_priority(PK)	需求变更优先级
ri_priority_name	名称
ri_priority_description	简述

表 20: 需求开发优先级表 ri\_priority

同时，我们需要将需求与需求开发进行绑定：

列名	含义
ri_id(PK)	需求开发识别符
r_id(PK)	需求表示符

表 21: 需求-需求开发关系表 r\_ri

## (2) 需求变更

设计表格如下：

列名	含义
c_id(PK)	需求变更识别符
c_reason	需求变更理由
c_description	需求变更描述
c_status(FK)	需求变更状态
c_time	变更提交时间
c_priority(FK)	变更优先级

表 22: 需求变更表 change

需要对需求变更状态和优先级单独建立表格：

列名	含义
c_status(PK)	需求变更状态
c_status_name	名称
c_status_description	简述

表 23: 需求变更状态表 c\_status

列名	含义
c_priority(PK)	需求变更优先级
c_priority_name	名称
c_priority_description	简述

表 24: 需求变更优先级表 c\_priority

同时，我们需要将需求与需求变更进行绑定：

列名	含义
c_id(PK)	需求变更识别符
r_id(PK)	需求表示符

表 25: 需求-需求变更关系表 r\_change

## 5. 需求跟踪

需求跟踪可以仿照需求开发等设计表来进行设计。

列名	含义
rt_id(PK)	需求跟踪符
rt_implement	需求跟踪接口
rt_file	需求跟踪文件说明
rt_status(FK)	需求跟踪状态
start_time	需求跟踪开始时间
rt_priority(FK)	需求跟踪优先级

表 26: 需求跟踪表 requirement\_trace

需要对需求跟踪状态单独建立表格：

列名	含义
rt_status(PK)	需求跟踪状态
rt_status_name	名称
rt_status_description	简述

表 27: 需求跟踪状态表 rt\_status

需要对需求跟踪优先级单独建立表格：

列名	含义
rt_priority(PK)	需求跟踪级
rt_priority_name	名称
rt_priority_description	简述

表 28: 需求开发优先级表 rt\_priority

同时，我们需要将需求与需求跟踪进行绑定：

列名	含义
rt_id(PK)	需求跟踪识别符
r_id(PK)	需求表示符

表 29: 需求-需求跟踪关系表 r\_rt

## 四、 表格分析

### 1. 表格分析之分类

前面一章一共设计了 29 张表格，为了进行数据库的建模和对业务的操作，我们需要进行整理归类。下面我们将设计出来的表进行分类整理，整理了以下几类：

#### (1) 业务实体类

业务实体类是指完成核心业务的实体类，包含了以下表格：

- (1) 需求来源信息表 requirement\_s: 主要定义了需求来源的各个属性，是需求导出的核心设计表格。
- (2) 需求定义表 requirement: 主要定义了需求的各个属性，是需求定义的核心设计表格。
- (3) 角色表 role: 主要定义了角色的各个属性，是需求分析之角色的核心设计表格。
- (4) 需求关系定义表 requirement\_relation\_define: 主要定义了需求关系的各个属性，是需求分析之需求关系的核心设计表格。
- (5) 需求确定表 validation: 主要定义了需求确定的各个属性，是需求确定的核心设计表格。
- (6) 需求开发表 requirement\_implement: 主要定义了需求开发的各个属性，是需求开发的核心设计表格。
- (7) 需求变更表 change: 主要定义了需求变更的各个属性，是需求变更的核心设计表格。
- (8) 需求跟踪表 requirement\_trace: 主要定义了需求跟踪的各个属性，是需求跟踪的核心设计表格。

#### (2) 减少冗余类

减少冗余类是指为了减少数据库冗余，单独抽象出来的属性表格，但又不占据业务的核心功能。减少冗余类的最大特点就是每一个类都大幅减少了数据库冗余，同时，虽然该类中的属性较少，但还也均为关键属性。由于该类较多，不再一一列举出来：

- 表 4: 将调查方法单独定义。
- 表 7: 将需求定义单独定义。
- 表 8: 将需求状态单独定义。
- .....

#### (3) 关系类

为了表示各个表之间的关系，比如 N:1,N:M,1:1 等等，尤其是为了表示 N:M 关系，我们需要解耦，单独定义关系表格，主要（未完全列举）包含了以下：

- (1) 表 17: 需求和需求确定有时候会极其复杂，关系为 N:M。
- (2) 表 21: 需求和需求开发有时候会极其复杂，关系为 N:M。
- (3) 表 25: 需求和需求变更有时候会极其复杂，关系为 N:M。
- (4) 表 29: 需求和需求跟踪有时候会极其复杂，关系为 N:M。

## 2. 表格分析之属性类型确定

我设计了 29 个表，但是，我只定义了含义，而没有定义属性类型，比如 char, varchar, int 等等，在此我不给出定义，主要是出于两方面考虑：

1. 属性类型往往是根据实际业务确定的，不同场景属性必然不同，不能一概而言，在本实验中思考属性类型不仅费时费力，而且毫无意义。
2. 属性类型的模糊性不影响我们需求分析业务的实现。

## 五、 数据库设计

在数据库设计中，为了表达清楚和操作方便，我将只设计业务实体和部分表格之间的交互关系，一些繁琐但简单的操作将不再给出，但并不代表不存在关系。

### 1. 需求导出

设计的数据库表格如下：

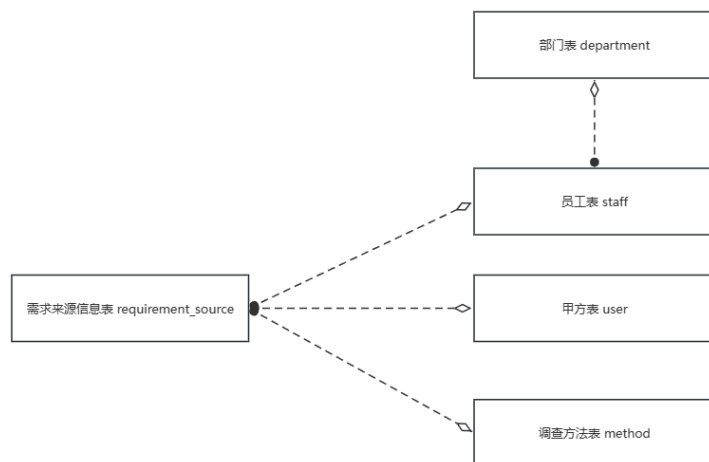


图 1: 需求导出

### 2. 需求挖掘

#### (1) 需求定义

设计的数据库表格如下：

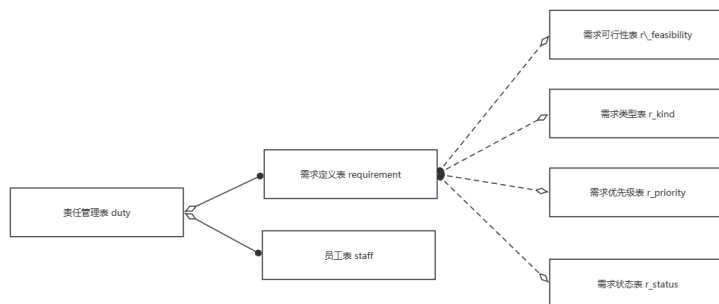


图 2: 需求定义

## (2) 需求分析之角色

设计的数据库表格如下：

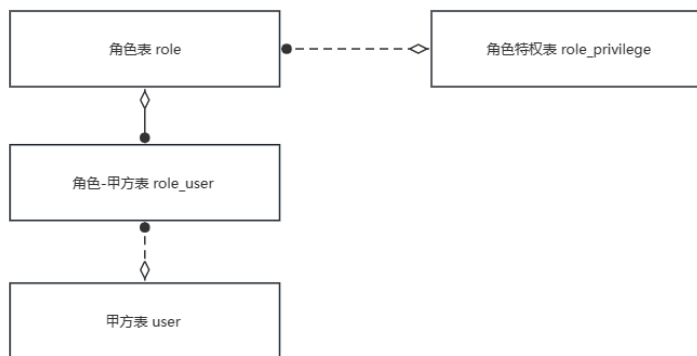


图 3: 需求分析之角色

## (3) 需求分析之需求关系

设计的数据库表格如下：

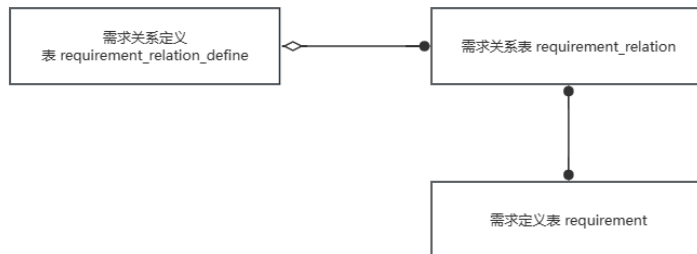


图 4: 需求分析之需求关系

### 3. 需求确定

设计的数据库表格如下：

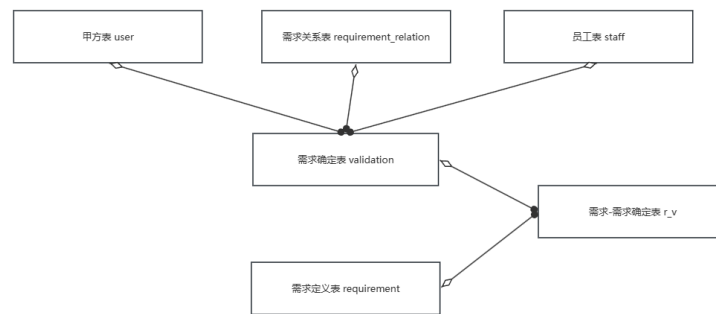


图 5: 需求确定

### 4. 需求开发及变更

#### (1) 需求开发

设计的数据库表格如下：

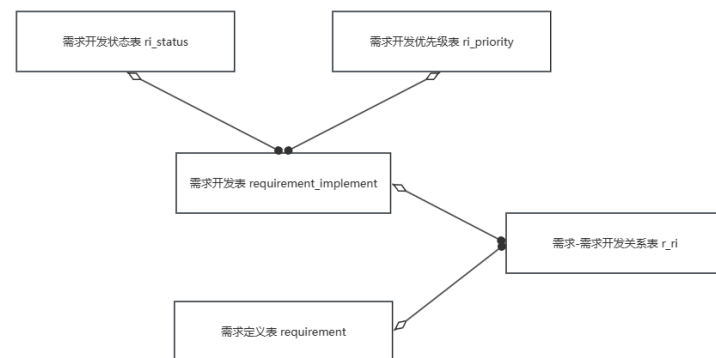


图 6: 需求开发

#### (2) 需求变更

设计的数据库表格如下：

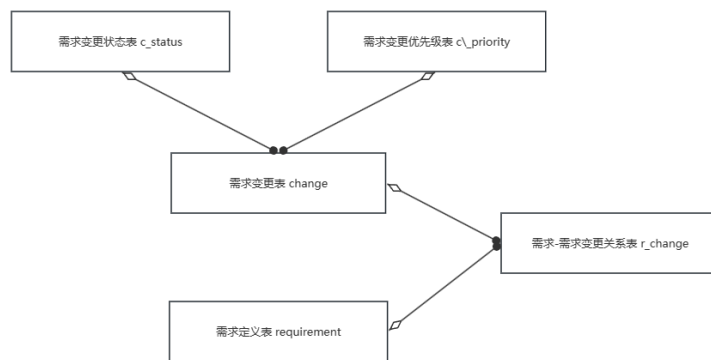


图 7: 需求变更

## 5. 需求跟踪

设计的数据库表格如下：

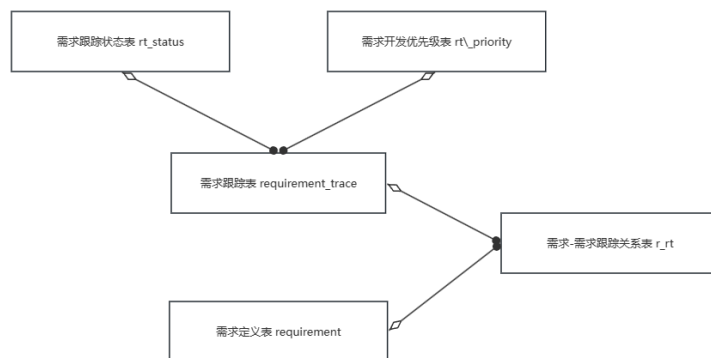


图 8: 需求跟踪

## 六、 实验总结

作为一名软件工程学生，需求管理系统数据库表设计是有趣而且锻炼能力的。在这个实验中，我将需求管理系统抽象为了 5 个部分：需求导出、需求挖掘、需求确定、需求开发和需求跟踪，并设计了相应的 SQL 数据库结构。

我不再赘述每个部分的设计思想了，下面我简单地总结我自认为设计的优点：

- (1) 充分地考虑实际工作场景，留足了弹性空间。比如有多个表设置了... 属性供未来工作人员选择填写。
- (2) 尽最大可能降低了数据库的冗余。
- (3) 充分考虑了需求的各个阶段，做出了各种变化的可能，比如需求变更，需求跟踪机制均不错。

以下为我认为的缺点：

- (1) 没有考虑非功能需求。

(2) 还是存在部分数据库冗余。

(3) 日志系统不完善。

总结：我回顾了每个部分的设计和实施过程，并讨论了数据库结构对于实现需求管理系统各个功能模块的有效性和可扩展性的影响。我还提及了在实验过程中遇到的挑战，并提出了改进建议或进一步的研究方向，以提高需求管理系统的性能和用户体验。

## 参考文献

- [1] IT 项目管理（原书第 8 版）Kathy Schwalbe
- [2] 软件需求管理系统的设计与分析 [D]. 吉林大学,2015. 孙乾.
- [3] <https://yijunquan.blog.csdn.net/article/details/122023770>