

## 独创性声明

本人郑重声明：所提交的学位论文是本人在导师指导下独立进行研究工作所取得的成果。据我所知，除了特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。对本人的研究做出重要贡献的个人和集体，均已在文中作了明确的说明。本声明的法律结果由本人承担。

学位论文作者签名： 王琳 日期： 2010.6

## 学位论文使用授权书

本学位论文作者完全了解东北师范大学有关保留、使用学位论文的规定，即：东北师范大学有权保留并向国家有关部门或机构送交学位论文的复印件和电子版，允许论文被查阅和借阅。本人授权东北师范大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编本学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名： 王琳 指导教师签名： 李林  
日 期： 2010.6 日 期： 2010.6

学位论文作者毕业后去向：

工作单位： \_\_\_\_\_ 电话： \_\_\_\_\_  
通讯地址： \_\_\_\_\_ 邮编： \_\_\_\_\_

1

2

3

4

## 摘 要

随着科技的进步和经济的繁荣，软件行业也在飞速的换代和发展。中国的软件行业也跟随着世界的步伐不断成熟并发展着。各个软件开发企业为了更好的管理软件开发的基本流程，以及员工的日常基本工作，开发配套使用了多种多样的软件项目管理系统。软件项目管理是为了使软件项目能够按照预定的成本、进度、质量顺利完成，而对人员、产品、过程和项目进行分析和管理的活动。软件项目管理的根本目的是为了让软件项目尤其是大型项目的整个软件生命周期（从分析、设计、编码到测试、维护全过程）都能在管理者的控制之下，以预定成本按期，按质的完成软件交付用户使用。而研究软件项目管理为了从已有的成功或失败的案例中总结出能够指导今后开发的通用原则和方法。

本文所要做的就是开发一套适合中小型软件开发企业的软件项目管理系统。具体情况具体分析，从需求分析、概要设计、编码、测试及维护等各个方面对中小型企业的软件规范行开发进行管理，并能对企业员工基本信息、工作进度、企业分工等进行科学化的管理和汇总。使中小企业软件开发规范化、合理化。

本系统的一个创新就是对基于 SOA 思想的软件项目管理系统的合理化开发进行研究，通过将 SOA 思想与软件工程的流程紧密的结合起来，将面向服务的思想（SOA）应用到管理软件项目的开发流程中，通过软件项目管理系统把面向服务思想（SOA）应用到其所开发的每一个项目中，将基础设施和实现发生的改变所带来的影响降到最低限度，更快地整合现实，减少成本和增加重用。通过以松散耦合的方式公开业务服务，企业可以根据业务要求更轻松地使用和组合服务，可以更轻松地创建、修改和管理它来满足不同时期的需要。

**关键词：**软件项目管理系统、SOA、软件管理、软件开发流程

## **Abstract**

With technological progress and economic prosperity, the software industry is also rapidly replacing and developing with the pace of the world. In order to better manage the basic process of the software development as well as the work routine of employees, the various software companies develop and use a variety of Software Project Management systems. Software Project Management is an activity of analyzing and managing the staff, products, process and project so as to successfully bring the software project in accordance with a predetermined cost, schedule, and quality to a close. The fundamental purpose of Software Project Management is to make the life circle (the whole process from analysis, design, coding to testing, maintenance) of the software projects especially the large-scale projects under the control of the manager and deliver it with scheduled cost and quality to the software users. And Software Project Management summarizes the general principle and methods guiding the future development from the previous success or failure cases.

This thesis aims to develop a set of Software Project Management Systems for small and medium software development companies. It manages the software specification development of small and medium companies from the aspects of needs analysis outline design, coding, testing and maintenance and scientifically summarize the basic information for employees, work progress and company division enabling the software development of small and medium companies standardization and rationalization.

The innovation of the system is to study the rationalized development of SOA-based software project management. Through the close integration of SOA ideas and software engineering process, it applies the service-oriented concept (SOA) to the development process of the Software Project Management through Software Management System, it applies the service-oriented ideas (SOA) to each of its development projects to minimize the impact on the Changes from Infrastructure and realization and to quickly integrate the reality, and reduces costs and increase reuse. By opening business services in a loosely coupled manner, according to business, companies can easily use and combine services, and more easily create, modify and manage it to meet the needs of different periods of time.

**Key words:** Software Project Management System; SOA; software management; software development process

# 目 录

摘 要	I
Abstract	II
目 录	III
第一章 绪 论	1
1.1 研究背景	1
1.1.1 项目管理系统国内外研究状况	1
1.1.2 SOA 的应用现状	2
1.1.3 存在的问题	3
1.2 研究目标与内容	3
1.2.1 研究目标	3
1.2.2 研究内容	3
1.3 研究意义	4
第二章 相关技术的介绍	5
2.1 软件工程的相关介绍	5
2.1.1 软件工程的基本概念	5
2.1.2 软件工程的目标	5
2.1.3 依照软件工程理论开发软件的基本步骤	5
2.2 SOA 思想的相关介绍	6
2.2.1 SOA 的概述	6
2.2.2 SOA 的核心	7
2.2.3 SOA 的作用	7
2.2.4 SOA 的生命周期	8
2.3 系统开发方法概述	9
第三章 基于 SOA 的软件项目管理系统方法的概述	11
3.1 软件项目管理系统的基本工作	11
3.2 基于 SOA 的软件项目管理系统	12
3.2.1 与其他软件项目管理系统比较与创新	12
3.2.2 基于面向对象、面向组件、面向服务(SOA)方法的关系	12
3.2.3 SOA 软件系统服务的构建过程	13
3.2.4 SOA 软件系统服务的构建方法	14
3.3 应用 SOA 思想如何进行软件项目管理和开发方法的研究	15
3.3.1 SOA 的模块化的概念	15

3.3.2 如何应用 SOA 的思想管理软件项目的开发方法的研究.....	16
第四章 软件项目管理系统的设计与实现.....	19
4.1 系统的概述.....	19
4.2 环境的搭建.....	19
4.3 系统的功能设计.....	19
4.3.1 权限登录.....	20
4.3.2  workflow跟踪.....	21
4.3.3 建立项目.....	21
4.3.4 需求分析管理.....	21
4.3.5 概要设计管理.....	23
4.3.6 编码管理.....	26
4.3.7 基本信息管理.....	27
4.4 软件项目管理系统主要功能模块的划分与实现.....	27
4.4.1 登录及权限管理.....	28
4.4.2 人员管理.....	30
4.4.3 报表审批.....	32
4.4.4 项目维护.....	33
4.4.5 调研日志维护.....	37
4.4.6 文档维护.....	37
4.4.7 工时管理.....	39
4.5 系统数据库的设计.....	42
4.5.1 数据库语言的选取.....	43
4.5.2 数据库的连接.....	43
4.5.3 数据库表的设计.....	43
第五章 总结与展望.....	47
5.1 总结.....	47
5.2 展望.....	47
参考文献.....	49
致    谢.....	50

## 第一章 绪 论

### 1.1 研究背景

#### 1.1.1 项目管理系统国内外研究状况

##### (1) 国外:

在国外由于网络计划技术的出现,软件项目管理的发展可以追溯到五十年代中后期,网络分析程序到了六十年代中后期发展已经十分成熟。整个七十年代研究的重点是对网络模型分析软件的应用功能扩展和完善,如平衡优化其成本和资源;同时提出项目管理信息系统的概念,并对其进行研究<sup>[2]</sup>。进入八十年代以后,由于项目管理工作的标准化、科学化和计算机的普及,一般中小型企业以及中小型项目也可用计算机进行管理,网络技术才真正普及。九十年代后,项目管理软件发展十分迅速,并不断有使用方便、功能强大的软件推出,在项目管理中起到了重要作用<sup>[2]</sup>。

##### (2) 国内:

在二十世纪后期,国内对项目管理软件的研究才刚刚起步,至今为止大致发生了两次重大转变:

第一次是在九十年代初期,研发主体由用户本身转变为专业化的软件企业是这个时期的标志。在七十、八十年代多是由一些单位自行组织研制的比较简单化的管理,即个用户单位各自提出自己的需求并研究、开发、使用,是一种完全的小生产方式,在近二十年中缓慢的发展<sup>[1]</sup>。九十年代初,市场经济带动出几十家专门从事建筑管理软件开发的高科技民营企业,软件开始走上商业化、社会化、专业化的发展较为快速的道路。九十年代是国内管理软件迅猛发展的十年,涌现了如钢筋配料、标书制作软件、工程造价、平面图制作、工程量计算等新产品,这使得界面友好、操作方便,价格逐渐降低、功能不断完善,实用性、通用性增强<sup>[1]</sup>。

转变的第二次是在九十年代末,如将项目施工方案的设计、工程量计算、概预算、资源计划、进度计划、事务性管理、费用管理等综合起来形成一个有机的整体。由单机版转向系统集成是产品的标志。运行环境由单机用户拓展到网络多用户并实现了在某种程度上企业内部的数据共享<sup>[1]</sup>。

进入二十一世纪以后,研发项目管理软件的情况出现了滞缓现象。其中有多方面的原因,既有技术因素又有市场因素,既有自身条件的制约又有外部环境的影响,如加入 WTO 后产生的影响、用户需求的变化,数据库技术、通讯技术的进步、网络技术等。这些问题摆在众多业内人士面前了要面对的问题,如项目管理软件发展的方向以及发展的必然趋势,并且如何要如何应对这些问题<sup>[1]</sup>。

### 1.1.2 SOA 应用现状

面向服务体系结构 (SOA) 在全球范围内正成为企业未来的软件架构的趋势。埃森哲公司和《信息周刊》通过对中国上千家企业机构的问卷调查,了解在商业科技推动企业竞争力及商业创新的背景下,中国企业部署 SOA 的现状和未来的趋势<sup>[3]</sup>。

基于调研数据分析,可以发现 SOA 在中国的应用现状:

基础应用:

被调查企业中,拥有 SOA 应用的企业占 8.8%,在大中型企业中,这个比例是 10%。四分之三的被调查企业未对 SOA 采取任何行动。然而在较早开始 SOA 部署的企业中,60%的大中型企业和超过一半的小型企业都表示他们将增加在 SOA 上的投入。数据表明尽管 SOA 在中国尚处于应用启蒙阶段,但企业在 SOA 上的投入却在快速增加<sup>[3]</sup>。

91.2%的被调查小企业与 76.9%的大中型企业缺乏有效的应用系统管理制度,所有被调查企业中,仅有 15.5%企业拥有比较完善的应用管理制度。<sup>[3]</sup>

SOA 部署:

在了解 SOA 的被调查企业中,有 80.9%的企业表示将采取积极态度部署 SOA。已经或计划部署 SOA 的被调查企业,既包括大中型企业也涉及规模较小的企业,这反映出是否部署 SOA 与企业规模的联系不大,同时也表明通过部署 SOA,小企业可以缩小和大型企业之间的差距<sup>[3]</sup>。

价值和障碍:

有 88.8%的被调查企业优先认可 SOA 的商业价值。参与调查的大中型企业中,分别有 43.1%和 44.6%期望通过部署 SOA 促进企业的业务发展及流程优化。只有 12.3%的大中型企业认为 SOA 的价值主要是技术创新。以上数据表明部署 SOA 是一项业务驱动的组织变革,而不是技术驱动成本削减<sup>[3]</sup>。

整合现有应用系统 (40.6%) 以及整合业务流程 (38.2%) 是企业部署 SOA 的首要驱动因素。小企业对商业创新比较关注,15.8%的小企业认为,部署 SOA 是建立新的业务能力的基础。同时,38%的大中型企业也表示,建立新的业务能力是企业部署 SOA 在未来最重要的驱动因素。调查显示 90%可以支持现有业务需求的应用无法满足企业未来的业务发展,这对于未来的变化是一个巨大的推动力量,而 SOA,由于其能有效加强现有应用的能力,被视为在这方面投入的优先选择<sup>[3]</sup>。

中国企业部署 SOA 的主要困难除了在于流程再造的准备不充分 (42%) 以及无法明确 SOA 的商业价值 (41.9%),44%的大中型企业还强调了与管理层沟通的困难<sup>[3]</sup>。

策略:

在 SOA 实施的不同着手点中,28.7%的被调查企业选择测试单个 SOA 应用、33.8%选择基于 SOA 的系统及信息集成、19.3%选择基于 SOA 的部门内部自动化、13.4%选择跨部门的 SOA 应用、4.8%选择基于 SOA 的合作伙伴产业链。对于如何部署 SOA 的分歧意见显示 SOA 的部署并没有统一的方法,有些企业可能通过业务流程的驱动来实现,而另一些企业则可能通过对技术的集中变革实现。因此,一张清晰的,为企业量身订制的 SOA 部署路线图将是避免投资浪费的最佳选择。四分之三的被调查企业希望在部

署 SOA 过程中借助外部资源<sup>[3]</sup>。

### 1.1.3 存在的问题

通过对国内外正在使用的软件项目管理系统的分析可以得出，面向服务的思想（SOA）很少被应用到软件项目管理的开发流程中。即使某些系统中应用到一些 SOA 的思想去管理软件开发流程，但还没有使用 SOA 的思想去进行整体的规划和设计。对整个项目面向服务的效果也没更深入的评估。

以上情况导致系统所管理的项目开发周期相对较长，软件重用性低、大大增加了开发成本。

## 1.2 研究目标与内容

### 1.2.1 研究目标

通过基于面向服务（SOA）的软件项目管理系统的研究，将 SOA 与软件工程的思想紧密结合起来。将 SOA 的思想应用到软件项目的开发流程中。并通过软件项目管理系统把面向服务思想（SOA）应用到所开发的每一个项目中。

### 1.2.2 研究内容

#### (1) 应用 SOA 思想进行软件项目管理的研究

将面向服务的思想（SOA）应用到软件项目管理开发流程中，使得在项目的整个生命周期应用 SOA 的思想去进行整体的规划和设计。

#### (2) 基于 SOA 的软件项目管理系统的开发

本系统的目的是建立一套完善的项目开发 workflow 管理系统。通过人与计算机共同工作的自动化协调、控制和通讯，在计算机化的业务过程上，使所有命令的执行都处于受控状态。在项目开发的过程中，项目参与人员的具体工作情况可以通过时序图的方式体现出来，工作人员每天的工作完成情况以及遇到的技术难题解决方法都会以工作日志形式记录下来，既方便个人对工作的总结，又方便管理层的管理。

同时，从项目确立开始的所有工作都会以文档的形式记录并存档，以备查询。总之，这一系统可以有效的帮助管理人员管理整个工作流程，提高软件开发的工作效率。

项目开发的过程中，项目参与人员的具体工作情况可以通过时序图的方式体现出来，工作人员每天的工作完成情况以及遇到的技术难题解决方法都会以工作日志形式记录下来，既方便个人对工作的总结，又方便管理层的管理。从项目确立开始的所有工作都会以文档的形式记录并存档，以备查询。

本系统一个创新点就是将 SOA 这种思想应用到软件项目开发管理的体系中，简单的说通过 SOA 思想在本系统中的应用，使在本系统管理下的软件开发流程遵循 SOA 的规则，增强的所管理开发软件的面向服务的特性，增加代码的重用性，使所开发的项目可以跨语言，跨平台的结合。以便大大缩短的软件开发的工期，使软件开发面向服务的逻辑性大大加强。

### 1.3 研究意义

(1) 用软件项目管理系统去管理软件开发的意義：

软件项目管理系统在软件开发公司使用的含义重大，在实施层，项目管理软件是项目经理和项目团队执行项目的工具。在管理层，项目管理软件是固化企业级项目管理体系的工具。在决策层，项目管理软件是项目执行透明化、掌握数据科学决策的支撑工具。我们的软件项目管理软件正是以实现项目管理的“过程透明化”和提升组织级项目管理能力为理念而设计的。我们的软件项目管理软件遵循 PMI 的项目级和组织级管理的理论模型，覆盖项目—项目群—项目组合的管理，以及“项目管理环境”的管理。项目管理不是在真空中进行的，项目级管理需要组织内部环境的支持，即组织级管理的支持。

(2) 将 SOA 的思想应用到管理软件项目的开发流程中的意义：

将 SOA 的思想应用到软件项目管理系统中，不仅能体现软件项目管理系统管理软件开发的優勢，也可将 SOA 的思想贯穿到整个软件开发流程中。将面向服务的思想（SOA）应用于软件项目管理的开发流程，可使集成和管理变得更为容易，大大减少了代码的重用，缩短了软件的开发周期。

## 第二章 相关技术的介绍

### 2.1 软件工程的相关介绍

#### 2.1.1 软件工程的基本概念

软件工程 (Software Engineering, 简称为 SE) 是一门研究用工程化方法构建和维护有效的、实用的和高质量的软件的学科。它涉及到程序设计语言、数据库、软件开发工具、系统平台、标准、设计模式等方面。软件工程一直以来都缺乏一个统一的定义, 很多学者, 组织机构都分别给出了自己的定义:

Boehm: 运用现代科学技术知识来设计并构造计算机程序及为开发、运行和维护这些程序所必需的相关文件资料<sup>[4]</sup>。

IEEE: 软件工程是开发、运行、维护和修复软件的系统方法<sup>[4]</sup>。

Fritz Bauer: 建立并使用完善的工程化原则, 以较经济的手段获得能在实际机器上有效运行的可靠软件的一系列方法<sup>[4]</sup>。

#### 2.1.2 软件工程的目标

提高软件的质量与生产率是软件工程的基本目标, 实现软件的工业化生产是软件工程的最终的目标。软件需求方最关心的问题则是软件质量, 物美价廉即使不是用户的基本要求, 但货真价实也是用户的必要保证。软件供应方最关心的是生产率, 用更少的时间挣更多的钱是软件供应方的基本思想。质量与生产率之间存在着内在的联系, 高生产率必须把质量合格当作最主要的前提。如果达不到合格的质量, 供需双方都是难以接受的。从短期效益看, 追求高质量会延长软件开发时间并且增大费用, 看起来似乎降低了生产率。但从长期效益看, 高质量是软件开发的全过程更加规范流畅的保证, 从而大大降低了软件的维护代价, 这样是从实质上提高了生产率, 同时获得企业生存的保证—信誉。所以质量与生产率之间根本的对立是不存在的, 好的软件工程方法可以同时提高质量与生产率。在成本、进度的给定的前提下, 开发出具有可重用性、可修改性、可适应性、可理解性、可移植性、可维护性、可追踪性和可互操作性并且满足用户需求的软件产品<sup>[5]</sup>。

#### 2.1.3 依照软件工程理论开发软件的基本步骤

依照软件工程理论开发软件的流程主要包含了需求分析、概要设计、详细设计、编码和测试五个阶段。

需求分析: 需求分析是指理解用户需求, 在软件功能等方面与客户达成一致, 对软件风险和项目代价进行评估, 并且最终形成开发计划的一个复杂过程。在这个过程

中,处在主导地位的的确使用户,而需求分析工程师和项目经理要的职责是整理用户需求,为以后的软件设计打下基础<sup>[6]</sup>。

需求分析从广义上理解:需求分析包括需求的获取、分析、规格说明、变更、验证、管理的一系列需求工程。

需求分析狭义上理解:需求分析指需求的分析、定义过程。

概要设计:概要设计的主要任务是将需求分析得到的 DFD 转换为软件结构和数据结构。设计软件结构的具体任务是:将一个复杂系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及人机界面等。数据结构设计包括数据特征的描述、确定数据的结构特性、以及数据库的设计。显然,总体设计建立的是目标系统的逻辑模型,与计算机无关<sup>[7]</sup>。

详细设计:详细设计的主要任务是对每个模块的实现算法进行设计、对所需的局部数据结构进行设计。详细设计有两个目标:第一个是实现模块功能的算法在逻辑是正确的并且对算法的描述的基本要求是要简明易懂。

结构化程序设计法是传统软件开发方法的详细设计的主要方法。详细设计的表示工具有两种,分别是图形工具和语言工具。图形工具有程序流程图、PAD (Problem Analysis Diagram) 图、NS (由 Nassi 和 Shneidermen 开发,简称 NS) 图。语言工具有伪码和 PDL (Program Design Language) 等。

编码:编码阶段主要工作是依照软件工程中概要设计以及详细设计中项目的具体设计,进行软件项目代码的编写,完成项目的具体设计。

测试:软件测试就是利用测试工具按照测试方案和流程对产品进行功能和性能测试,甚至根据需要编写不同的测试工具,设计和维护测试系统,对测试方案可能出现的问题进行分析和评估。执行测试用例后,需要跟踪故障,以确保开发的产品适合需求。软件测试主要工作内容是验证和确认,下面分别给出其概念:验证是保证软件正确地实现了一些特定功能的一系列活动,即保证软件做了你所期望的事情。确认是一系列的活动和过程,目的是想证实一个给定的外部环境中软件的逻辑正确性。即保证软件以正确的方式来做了这个事件<sup>[8]</sup>。

## 2.2 SOA 思想的相关介绍

### 2.2.1 SOA 的概述

面向服务的体系结构(service-oriented architecture, SOA)是一个组件模型,它将应用程序的不同功能单元(称为服务)通过这些服务之间定义良好的接口和契约联系起来<sup>[9]</sup>。接口是采用中立的方式进行定义的,它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互<sup>[9]</sup>。

### 2.2.2 SOA 的核心

整个 SOA 实现的核心是服务,所谓的一组实体(服务提供者,服务消费者,服务注册表,服务条款,服务代理和服务契约)即可称为 SOA,如何提供和消费服务被这些实体详细说明了。所有的服务是模块化的、可互操作的、独立的、松散耦合、位置明确的并且可以通过网络查找其地址。服务是 SOA 架构风格中最核心的抽象手段,一系列粗粒度的业务服务和业务流程对业务进行划分。相对独立,自包含,可重用,由一个或多个分布的系统实现业务服务,而服务业把业务流程组装而来<sup>[10]</sup>。“服务”定义了一个与业务数据或业务功能相关的接口,并且包含约束这个接口的契约,如法律法规的遵循、服务质量的要求、安全性要求、业务规则等。接口和契约采用中立的形式,基于标准方式进行定义,接口和契约对于实现服务的硬件平台而言是相对独立的操作系统和编程语言。这样便使得可以以一种统一的和通用的方式进行交互,相互理解来构建在不同系统中的服务<sup>[10]</sup>。“服务”之间的位置是透明的,交互是动态的,技术和位置的透明性,使得服务的提供者和请求者之间高度解耦。这样使服务形成的松耦合系统的优点有两点:一点是当某个服务的内部结构和实现逐渐发生改变时,不影响其他服务;另一点是它适应变化的灵活性<sup>[11]</sup>。

### 2.2.3 SOA 的作用

需要使业务 IT 系统变得更加灵活,这是对 SOA 的需要的基本来源,这样就可以充分适应业务中的变化<sup>[12]</sup>。SOA 需求的可以通过允许强定义的关系和依然灵活的特定来实现,这样软件系统既可以运用现有系统的功能,又可以准备在以后做出一些改变,用来满足它们之间交互的需要。

举一个例子来进行具体的说明。例如一个服装零售组织拥在国际上拥有五百家连锁店,它们为了赶上时尚的潮流,必须时常更改设计<sup>[12]</sup>。这就意味着不仅需要进行颜色或样式的简单更改,甚至还可能需要对制造商、布料或可交付的产品进行更改。这是如果制造商和零售商之间的系统出现不兼容的情况,那么可能就会出现从一个供应商到另一个供应商的更换,这是一个非常复杂的软件流程。面对这种情况,就可以通过 WSDL 接口在操作方面的灵活性方面的利用,是的每个公司都可以保持它们的现有系统现状,而仅仅去匹配 WSDL 接口并制订新的服务级协定,这样就不必对软件系统进行重构了。这仅仅是对业务的水平进行变更,换句话说,它们改变的是合作伙伴,而所有的在业务上的操作基本都保持不变。这里,业务接口可以作少许更改,而内部的操作却不需要进行更改。另一种形式是在内部进行改变,在这种变化中,零售组织现在决定它还将把连锁零售商店内的一些地方出租给专卖流行衣服的小商店,这可以看作是采用店中店(store-in-store)的业务模型<sup>[12]</sup>。这里,虽然公司的大多数业务操作都保持不变,但是它们现在需要新的内部软件来处理这样的出租安排。尽管在内部软件系统可以承受全面的检修,但是它们需要在这样做的同时不会对与现有的供应商系统的交互产生大的影响。在这种情况下,SOA 模型保持原封不动,而内部实现却发生了变化。虽然可以将新的方面添加到 SOA 模型中来加入新的出租安排的责任,

但是正常的零售管理系统继续如往常一样。

项目经理也许会发现，软件新的配置还可以用另一种方式加以使用，以此来延续内部改变的观念，例如将粘贴海报的地方出租用来供广告之用。在这里，在新的设计中重用灵活的 SOA 模型才能得出新的业务提议。这不仅仅是 SOA 模型取得的一个新成果，并且也是一个新机会，而在以往有可能是不会出现这样的机会的。

这使得垂直改变也成为可能，在垂直改变中，零售商可以从销售他们自己的商品从而转变为专门通过 store-store 模型出租地方。如果完全从最底层开始垂直改变的话，这样就会使 SOA 模型结构发生显著改变，与此同时可能有新的系统、软件、流程以及关系与之同时发生改变<sup>[12]</sup>。在这种情况下，SOA 模型的优点是不是从应用程序和程序的角度考虑问题而是它从业务操作和流程的角度考虑问题，这样可以根据业务的操作清楚地确定什么需要添加、修改或删除。然后可以将为适合业务处理的方式来构造软件系统<sup>[12]</sup>。

在这里最重要的部分是改变和 SOA 系统适应改变的能力。对于软件的研发者来说，无论是在他们工作的范围之内还是在他们工作的范围之外都有可能发生这样的改变，这取决于是否有改变需要知道接口是如何定义的以及它们相互之间<sup>[12]</sup>。与软件的研发者不同的是，引起对 SOA 模型大的改变是架构师的职责。基于这样的分工，就是让软件的研发者集中精力对作为服务定义的功能单元进行创建，而让架构师和建模人员集中精力于将这些单元适当地组织在一起进行规划<sup>[12]</sup>。

SOA 对于面向同步和异步应用的，基于请求/响应模式的分布式计算来说可能会是一场革命。消费者或客户端被一个应用程序的业务逻辑 (business logic) 或某些单独的功能被模块化并作为服务所呈现。松耦合特性也是这些服务的关键<sup>[12]</sup>。例如相互独立服务的接口和实现。通过组合一个或多个服务可以被应用开发人员或者系统集成者来构建应用，而无须对服务的底层实现进行理解。例如，一个服务可以用 J2EE 或 .NET 来实现，除了使用的语言也可以不同外，使用该服务的应用程序可以在不同的平台之上。

#### 2.2.4 SOA 的生命周期

由于业务的诸多方面被 SOA 所涉及到，正因为这样就需要对 SOA 项目从一开始就进行细心的设计和规划。这样整个项目的生命周期都需要被考虑，从项目最初的阶段到第一个实现，再一直到有可能发生的重用和修订<sup>[13]</sup>。

##### 建模

相关技术来与面向服务的体系结构项目的第一步几乎没有任何的联系，所有事项都与您的业务相关。业务所执行的活动被面向服务的方法视为服务，因此第一步是要对这些业务活动或流程实际是什么进行确定。并记录您的业务体系结构，这些记录的作用在于不仅可以对 SOA 进行规划，还可以对实际业务流程进行优化。您可以通过在编写代码前模拟或建模业务流程更深入地了解这些流程，这样便有利于构建并且帮助执行这些流程的软件项目<sup>[14]</sup>。

于预期实现的深度决定了建模业务流程的程度。不仅仅如此，这个程度还依赖于开发团队中担任的角色。例如您是企业软件架构师，对实际的业务服务进行建模是您的主要工作。如果您是软件研发人员，对单个服务进行建模可能会是您的主要工作。

### 组装

开发人员可以在对业务流程进行了建模和优化后开始构建新的服务和/或重用现有的服务，随后对其进行组装，从而形成组合应用程序，以便实现这些流程。您在建模步骤中已经确定了已经存在某种形式的实现这些服务或访问该类数据所需的一些软件并且需要何种类型的服务以及它们将访问何种类型的数据<sup>[14]</sup>。找到已经存在的功能，并为其添加服务支持是“组装”步骤的主要工作。除此之外，为了满足您的 SOA 涉及的业务流程范围内的需求，还涉及到创建提供功能和访问数据源所需的新服务。

### 部署

在进行了建模、组装两个步骤后，要对组成 SOA 的资产进行部署，将其部署到安全的集成环境中。这种环境本身是为了集成业务中涉及的人员、流程和信息而提供专门化的服务。为了确保将公司的所有主要元素连接到一起协同工作，必须需要这种级别的集成<sup>[14]</sup>。此外，满足业务的性能和可用性需求是部署工作主要任务，部署工作还要提供足够的灵活性，用来吸纳新服务（并使以前的服务退役），而不会对整个系统产生过多的影响<sup>[14]</sup>。

### 管理

在一切都正常运行，系统就位后也不可以对一切放手不管。在部署后，需要从业务和 IT 这两个角度对您的系统进行监视和管理。收集的信息用于帮助实时地了解业务流程，从而能更好地进行业务决策，并将信息反馈回生命周期，以进行持续的流程改进工作是在“管理”步骤中的主要工作。接下来需要处理安全、服务质量、一般系统管理之类的工作。

### 控制

SOA 是一种集中系统；其中可以包含来自组织的不同部门的服务，甚至还能包含来自组织外的服务。如果没有恰当的控制，这种系统很容易失控。

控制对所有生命周期阶段起到巩固支撑作用，为整个 SOA 系统提供指导，并有助于了解系统全貌。它提供指导和控制，帮助服务提供者 and 使用者避免遇到意外情况。

## 2.3 系统开发方法概述

本软件项目管理系统是在 Jboss Seam 框架基础上融合了 SOA 思想以及 JSF 工作流跟踪原理设计的软件项目管理系统。

Seam 是一个 Java EE 5 框架。它通过把 JSF 与 EJB3.0 组件合并在一起，从而为开发基于 Web 的企业应用程序提供一个最新的模式。Seam 可以让你把 EJB 组件直接绑定到 JSF 页面。Seam 还可帮助你把 JBPM 流程定义直接地集成到你的应用程序中

<sup>[15]</sup>。

JSF 和 EJB 3.0 是 Java EE5 的两个最好的新特性。EJB3 是服务器端业务和持久逻辑的全新组件模型。同时，JSF 也是表现层的一个优秀组件模型。不幸的是，这二者都无法独自解决所有的计算问题<sup>[15]</sup>。实际上，JSF 和 EJB3 结合使用后运作得最好。但是 Java EE5 规范并没有提供如何整合这两个组件模型的标准方法。所幸，这两种模型的创建者都前瞻到了这种状况，并且提供了标准的扩展点，允许对各自进行扩展，或者与其他解决方案集成<sup>[15]</sup>。

Seam 将 JSF 和 EJB3 的组件模型合二为一，消除了胶合代码，使得开发者专注于业务问题。编写“一切”都是 EJB 的 Seam 应用程序是有可能的。如果你习惯把 EJB 当作是细粒度的所谓“重量化”的对象，这可能会令你很吃惊。然而，从开发人员的角度来看，3.0 版本已经完全改变了 EJB 的本质。EJB 是一个细粒度的对象—没有什么东西会比注解的 JavaBean 更复杂了。Seam 甚至鼓励你使用会话 Bean 作为 JSF 动作监听者<sup>[16]</sup>！

另一方面，如果你宁可不在这个时候采用 EJB 3.0，不用勉强。事实上，任何 Java 类都可以是一个 Seam 组件，并且 Seam 提供了你期待从“轻量化”的容器，甚至任何组件、EJB 或者其他东西中获得的所有功能<sup>[16]</sup>。

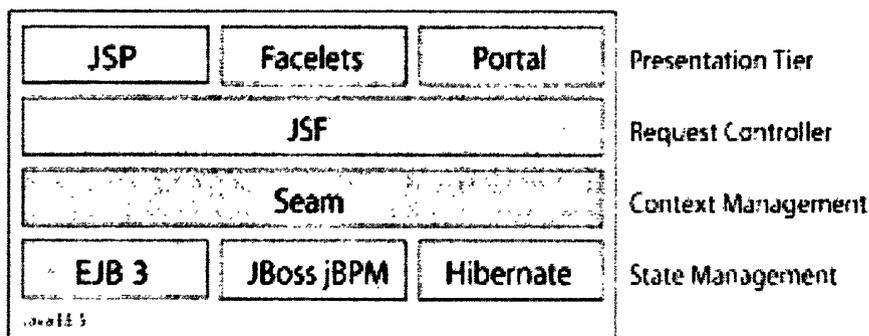


图 2.1 Seam 关系图

## 第三章 基于 SOA 的软件项目管理系统方法的概述

### 3.1 软件项目管理系统的基本工作

#### 1) 项目开发需要的资源

在二十一世纪初中国软件评测中心调查中发现,中小型软件项目中所需资源及其在项目成功中所占比例为:稳定的团队占 31%,明确的需求占 24%,有经验的管理者占 20%,充裕的时间占 13%,成熟的技术只占 10%。可见技术已经不是决定软件项目成败的最主要原因,而团队的稳定性和明确的需求是最重要的。所以软件管理系统最重要的职责就是保证团队的稳定以及确保需求的明确<sup>[16]</sup>。

#### 2) 安排项目开发计划

有明确的目标、明确的时间期限是安排任务要做的主要工作。如果目标不在最初的时候明确,就有可能会在完成的过程中受环境的影响而进行调整,最后可能会使目标发生偏移、控制不严格或时间期限不明确,这有可能会使工程无限期的延长。要明确划分任务,要在各任务执行之前就要对任务之间的关系明确的进行定义,要尽可能降低各模块任务耦合程度,这样才能将各块任务之间相互影响减到最小。另外,要在任务划分时还要作好局部重做的准备,所以对模块的划分和架构的设计来说,必须要能够满足局部重做的需要。

#### 3) 写好项目开发文档

要写好项目的开发文档是项很重要的工作。在软件开发过程中,需要注重细节非常多,如果稍不留神就会发生遗漏,所以要时时做好开发文档记录工作。要做好思路的整理过程,将我们需要记载的所有信息在文档中记录下来,进行查询、归纳、分类、整理、删减、补充等,然后整理出一套明确、完整、逻辑严密的思路,写好项目开发文档是必要的工作。沟通是写文档的最高的境界,要让别人清楚的理解你所写的东西,这件事是不简单的,要让别人看懂,不仅要思路清晰,逻辑严密,还需要有层次、有主次的论述观点。文档对沟通来说起到了至关重要的作用,在项目开发过程中,文档对开发的沟通起了至关重要的作用,做项目运行阶段,文档对项目维护更新也起了相当重要的作用。

#### 4) 做好汇报管理工作以及工作流程的跟踪

在软件开发过程中,对工作进程的管理是相当重要的,对软件开发进度规范合理的管理可以大幅度的提高软件的开发速度,保证软件的开发质量,增加软件公司的效益,而软件项目管理系统所要做的就是管理好软件开发人员的进度汇报工作,以及软件开发管理人员的工作流程跟踪工作<sup>[17]</sup>。

### 3.2 基于 SOA 的软件项目管理系统

#### 3.2.1 与其他软件项目管理系统的比较及创新

在对国内外正在使用的软件项目管理系统的调查和分析中，面向服务的思想（SOA）很少被应用到管理软件项目的开发流程中，即使在某些系统中应用到一些 SOA 的思想去管理软件开发流程，但还没有在项目的整个生命周期应用 SOA 的思想去整体规划和设计，对整个项目的面向服务的效果也没做深入的研究和评估。致使软件项目管理系统所管理的项目开发周期长，软件重用性低、大大增加了开发成本。

利用 SOA 的思想开发出一个功能比较齐全的，效率比较高的、能够对软件公司的项目开发流程进行实时监控的项目管理系统，从而缩短软件开发的周期，提高软件开发的效率。

#### 3.2.2 基于面向对象、面向组件、面向服务（SOA）方法的关系

本软件项目管理系统的创新之处是用面向服务的思想（SOA）去管理软件项目的开发，首先我们需要理解基于面向服务（SOA）的软件开发方法，其次才能良好的应用 SOA 思想管理各类软件的开发。

伴随着软件开发行业的迅猛发展，不可避免的牵涉到异构、分布式等复杂的特征环境，导致了项目开发周期长，软件重用性低、大大增加了开发成本。从而多种架构与性环境如（Seam, .NET）等应运而生，架构环境关注与业务开发，各种业务功能以组件形式发布到各类架构环境中，软件的开发级别也上升到组件级别。此阶段所采用的方法是面向组件的方法。随着企业级的迅猛发展和扩大，对减少软件的重用，异构的要求也越发突出出来，面向服务的思想（SOA）应运而生，它是基于面向对象和面向组件方法的升华。应用了 SOA 的思想去管理所开发的软件。使得能充分利用现有的资产，集成和管理复杂性变得更为容易。将基础设施和实现发生的改变所带来的影响降到最低限度，更快地整合现实。如图 3.1 所示：

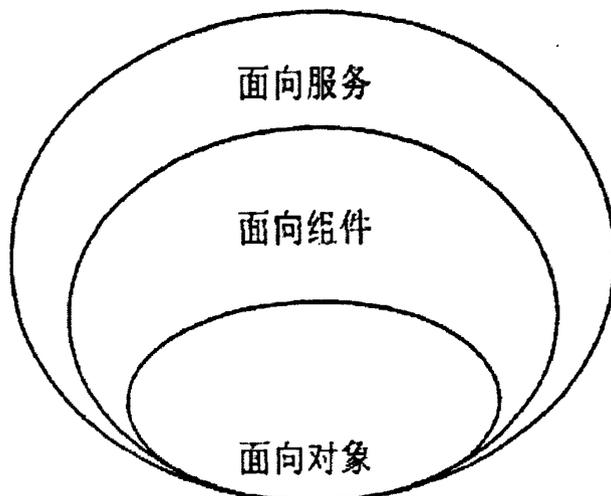


图 3.1 面向对象、组件、SOA 关系图

### 3.2.3 SOA 软件系统服务的构建过程

以往的面向对象、面向组件的设计方法的方法，总是先构建多个的应用，每个应用可能都有自己的功能层次、数据架构、安全架构等，然后随着系统的发展，需要在应用间进行整合，包括表示层整合、数据整合和流程整合等。根据客户的需求提取出业务流程或根据现有组件、系统提取出组件。从而应用面向对象、面向组件的设计方法进行软件开发设计。随着软件企业级的日益扩大，对减少软件的重用，异构的要求也越发突出出来。SOA 的出现，为软件组织构建灵活应对需求变化的软件系统指明了方向。它需要通过业务流程及组件中归提取出服务，然后将简单的服务编排成符合的服务，充分按照 SOA 思想进行软件的开发与设计<sup>[18]</sup>。

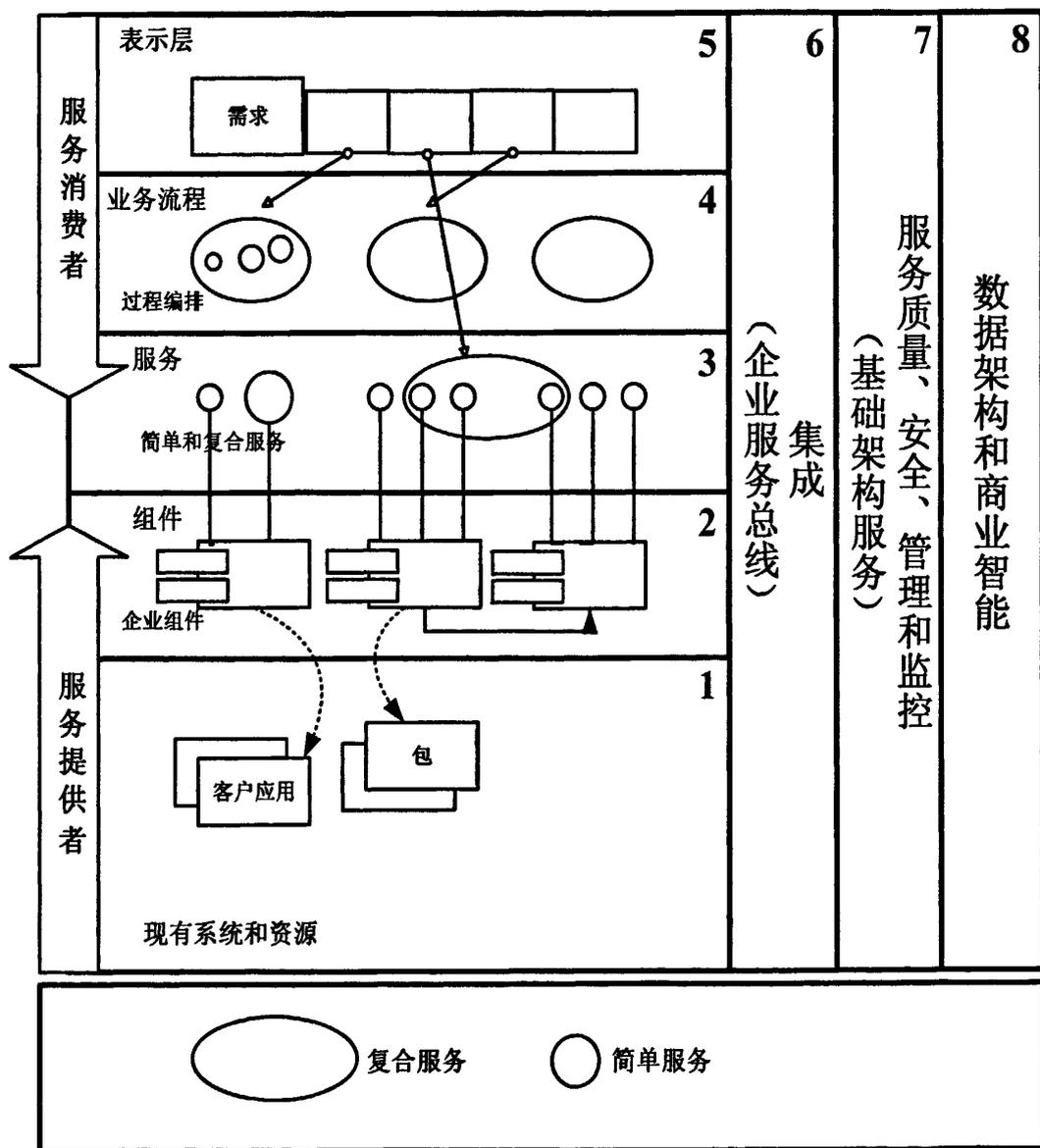


图 3.2 SOA 软件系统构建服务图

如图 3.2 所示，在服务消费者方面，系统开发人员在表示层系统需要做的是根据客户的需求，分析、归纳出整个系统的业务流程，形成业务流程模块。然后对业务流

程模块进行分析与编排。分别从功能，开发原理，开发代码等方面对业务流程进行编排，将功能，开发原理，开发代码等方面相似的业务流程进行提取，从而形成简单的服务。对简单服务进行分析，归纳，将相近的简单服务进行编排，形成简单或复合的服务，达到服务提取的目的。

从服务消费者方面来看，企业的资源，企业的组件需要和简单的服务或复合服务进行对应，如暂缺的企业组件资源，需要根据服务进行补足，最后形成一个个完整的服务包。从而达到客户的应用。以这种方式进行软件开发与设计，可以充分应用面向服务的思想（SOA）进行软系统的开发，减少软件的异构及重用。也可积累企业的资源，对软件项目的快速、规格化开发，以及企业的壮大都能起到而后很大的作用。

后续需要对企业的服务总线进行集成，对基础架构的服务进行服务质量、安全、的管理和监督，完整的进行数据架构和商业智能的应用。

### 3.2.4 SOA 软件系统服务的构建方法

服务是整个 SOA 实现的核心，如何进行 SOA 软件系统服务的构建，即如何根据需求和企业现有资源构造服务。将业务流程合理化归纳和编排成服务。构建服务的大致过程有初始、定义、设计、构造、部署、维护等几个步骤<sup>[19]</sup>，具体介绍如下：

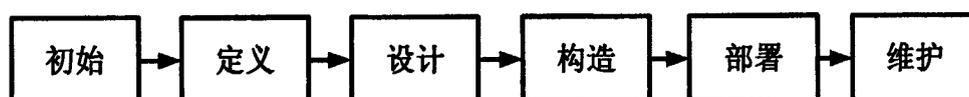


图 3.3 构建服务过程图

初始的基本工作：

在初始阶段主要需要完成的工作有：明确所要开发 SOA 软件项目的需求，明确开发目的、企业现有资源等。

定义的基本工作：

在定义阶段主要需要完成的工作有：对需求进行完整的收集与并进行合理化分析、非功能性需求定义、定义技术基础设施、项目计划制定、定义和实现用例、定义和记录总的体系机构等。需求分析与采用其他方法开发软件的过程和方法是相同的，首先要分析系统的功能性需求和非功能性需求。需要注意的是，分析系统功能时，不能像传统的方法那样按照各个部门来分析，而是需要按照业务流程来进行分析，按照业务流程定义出完整的服务。

设计的基本工作

在设计阶段主要需要完成的工作有：明确需求，对业务流程进行细化，并设计，编排出简单服务，为从功能，设计原理等方面联系紧密的简单服务设计出相应的符合服务<sup>[20]</sup>。

构造的基本工作

在构造阶段主要需要完成的工作有：依照需求分析，合理的 SOA 构建方法构造

服务并完成编码、测试等工作，对相关必要文档进行编写及保存。

#### 部署的基本工作

在部署阶段主要需要完成的主要有：项目开发人员对服务进行完整的部署，完成相关模型，及系统的开发工作。

#### 维护的基本工作

在维护阶段主要需要完成的工作有：系统维护人员对 SOA 系统进行必要的代码，功能等后续服务如错误纠正，根据新需求添加新功能等。

### 3.3 应用 SOA 思想如何进行软件项目管理和开发方法的研究

#### 3.3.1 SOA 的模块化的概念

服务支持的一个包含内聚性接口的接口集，即使在同一个模块下它们也应该彼此相关联<sup>[21]</sup>。模块化的原则应该在服务设计中被坚持，这样可以通过很少的依赖关系对一个或多个服务进行组合。下面列举了一些模块化的准则：

##### 一、模块的可分解性

每个应用可以被分解成许多小模块，其中每个小模块可以代表应用中一个单独的、不同的功能即是模块的可分解性。一个大的应用在自顶向下的设计中被分解成小的应用。可重用是分解模块的主要目的，即在不同的环境中对模块进行重用<sup>[22]</sup>。

##### 二、模块的可组合性

服务组件可以作为一个整体与其它服务自由地组合成新的服务系统即是模块的可组合型。服务设计者应该充分的对服务的独立性进行考虑，从而在完全不同的系统中对模块进行重用<sup>[22]</sup>。

##### 三、模块的可理解性

用户在不了解其它服务的情况下理解当前服务功能的能力即服务的可理解性。服务的一个至关重要的方面就是模块的可理解性，因为任何服务可以被服务消费者在任何时间查找并使用，如果功能方面的可理解性不被服务模块所具备，该应用则很难被用户决定是否使用<sup>[22]</sup>。

##### 四、模块的连续性

某个服务的改变将影响其它服务或服务消费者即服务的连续性。没有充分隐藏服务实现细节的接口在需要改变时将引起多米诺效应，在服务内部实现改变时要求其它服务和应用也要作相应的改变，因此每个服务都必须隐藏其内部设计实现的具体信息<sup>[22]</sup>。

##### 五、模块具有自保护性

服务中的异常条件不会级联影响其它的服务以及服务消费者即服务的自保护性。其它服务、用户、服务消费者之间的契约服务和内部数据状态不能被操作中的错误所影响<sup>[22]</sup>。

### 3.3.2 如何应用 SOA 的思想管理软件项目的开发方法的研究

在了解 SOA 软件系统服务的构建过程和 SOA 的构建方法后，我们要做的是如何应用面向服务（SOA）的思想去管理软件项目的开发与设计，也就是要研究如何应用所开发的软件项目管理系统去按照 SOA 软件系统服务的构建过程和 SOA 的构建方法去管理企业所要开发的软件，用软件项目管理系统去管理并约束企业按照 SOA 面向服务的思想去开发、管理、维护软件项目的开发<sup>[23]</sup>。

软件项目管理系统如果想要应用 SOA 的思想去管理软件项目的开发就要将软件工程、项目管理系统、SOA 思想的架构原则和过程紧密、有机的集合起来。

依照软件工程理论开发软件主要包含需求分析、概要设计、详细设计、编码和测试五个阶段<sup>[24]</sup>。在每个阶段完成各自的管理功能，完成对面向服务（SOA）思想的架构的管理，对 SOA 的构建过程即初始、定义、设计、构造、部署、维护进行全面的的管理。使项目管理系统完整的对开发软件项目进行全面的的管理，实现在软件开发全过程中遵循面向服务（SOA）的思想。

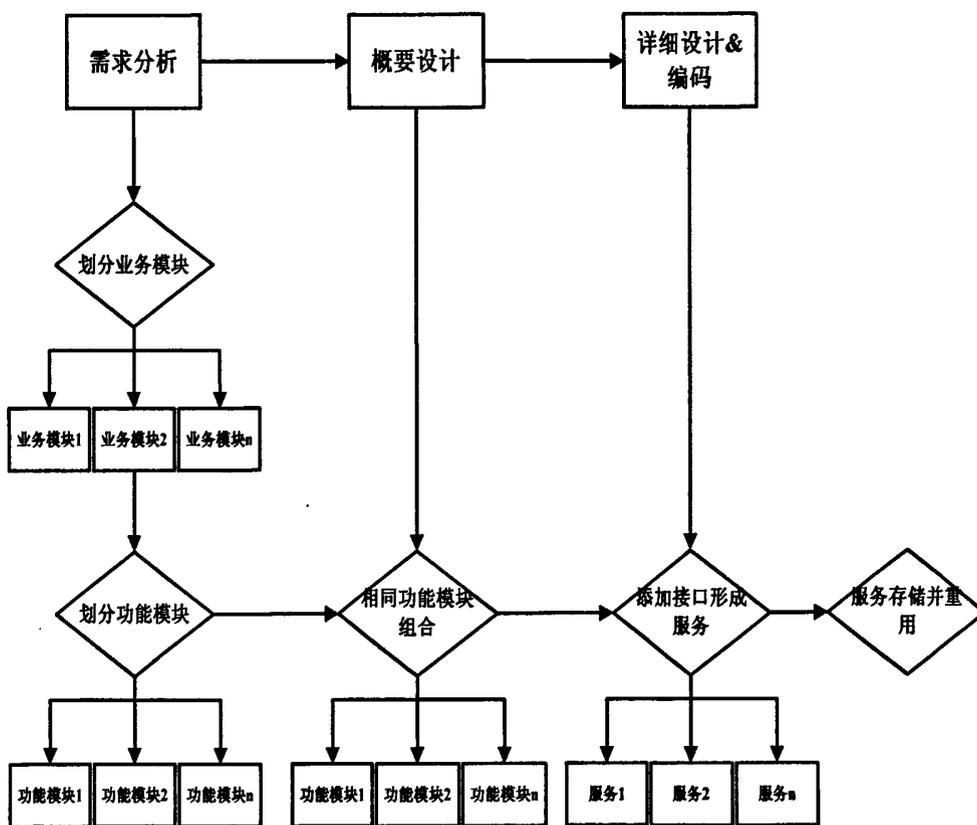


图 3.4 应用 SOA 思想管理软件项目的开发图

需求分析阶段：软件项目管理系统如果要应用 SOA 的模块化原理去管理软件项目，就得从软件开的需求分析做起。在以往的软件项目管理系统中通常只对业务模块进行划分并管理，而这里我们要做到对业务模块进行细分，实现模块的分解性。在自顶向下的设计中将一个大的应用分解成小的应用。首先将业务流程划分成一个个的

业务模块，对业务模块进行整理、分析，并进行功能模块的划分。在需求分析中主要应用到了面向对象、面向组件的思想，本软件项目管理系统在需求分析中主要的职责是管理好业务流的划分和功能模块的整理<sup>[26]</sup>。如图 3.5 所示：

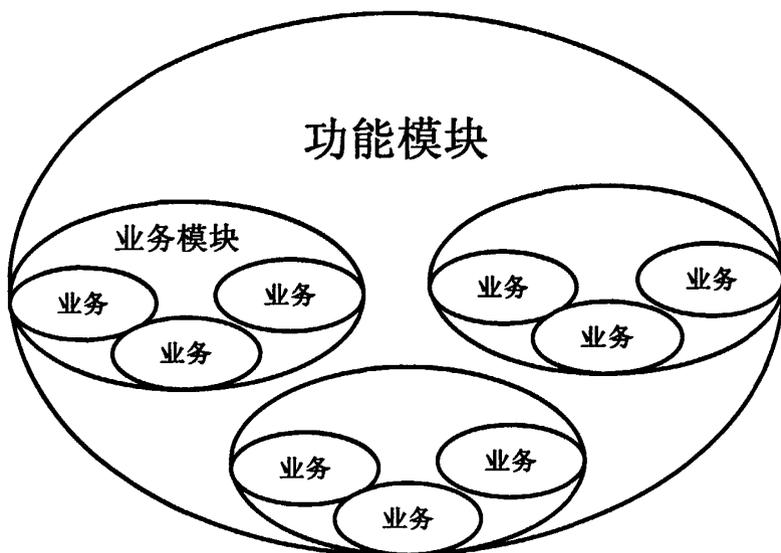


图 3.5 需求分析阶段基本工作图

概要设计阶段：在软件的概要设计阶段，除了要以往软件工程中概要设计基本工作进行必要的管理，而且还要对功能模块进行划分归类，将功能相同或相似的模块进行组合、归纳、提取，形成简单服务、并对相同或相似的简单服务进行组合，形成复合服务。本软件项目管理系统在这部分主要的功能为进行软件工程中概要设计流程进行必要的管理，并对简单服务、复合服务进行管理<sup>[26]</sup>。如图 3.6 所示：

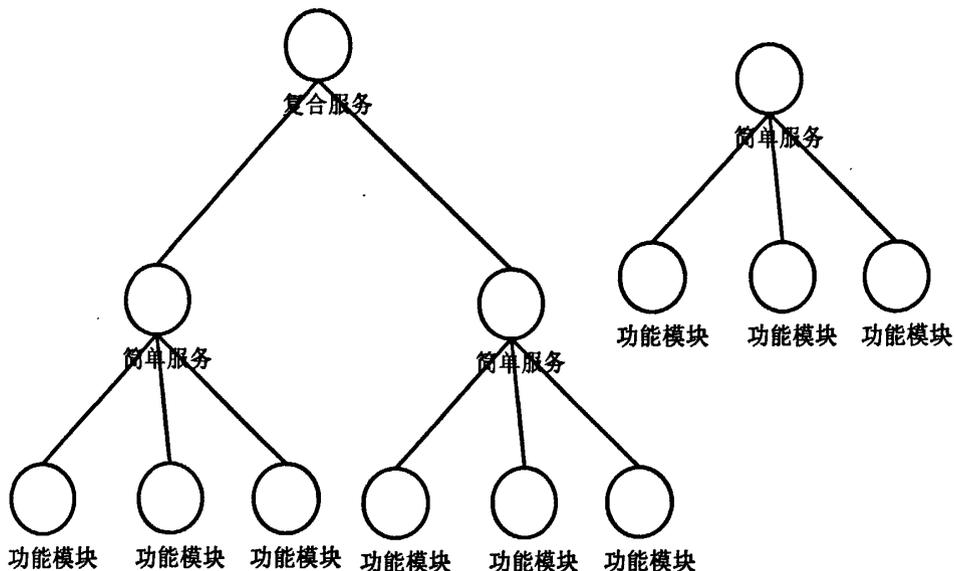


图 3.6 概要设计阶段基本工作图

详细设计和编码阶段：在软件的详细设计和编码阶段，除了要以往软件工程中详细设计基本工作进行必要的管理，还要完成详细设计，和编码的管理工作，完成所设计和归纳的服务的部署工作<sup>[27]</sup>。

测试和维护阶段：在软件的测试和维护阶段，除了要对以往软件工程中软件的测试和维护阶段基本工作进行必要的管理，还要对 SOA 系统进行必要的代码，功能等后续服务如错误纠正，根据新需求添加新功能等操作进行管理，完成 SOA 的后续维护工作。

总之，本软件项目管理系统除了要有管理软件开发的基本功能外，还要在软件开发的需求分析阶段对 SOA 的初始和定义进行管理，在软件的概要设计阶段对 SOA 的设计和构造进行管理，在软件的详细设计和编码阶段对 SOA 的部署进行管理，在软件的测试和维护阶段对 SOA 的后续维护进行管理。这样本软件项目管理系统就充分的应用面向服务（SOA）的思想对所开发的软件进行管理，使所管理开发的软件项目充分应用面向服务（SOA）的思想。

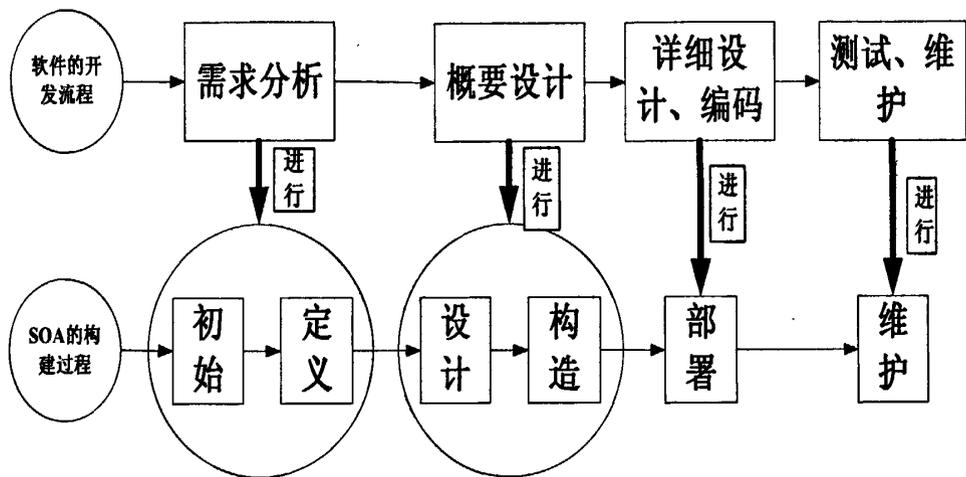


图 3.7 应用 SOA 思想管理软件项目的基本工作图

将 SOA 的开发思想应用到软件项目管理系统中，不仅可以体现用软件项目管理系统去管理软件开发的优点，也可将 SOA 的开发思想从软件开发开始应用到整个软件开发流程中。这样使得软件工程的流程可以完全的遵循 SOA 的思想。使得能充分利用现有的资产，集成和管理复杂性变得更为容易。将基础设施和实现发生的改变所带来的影响降到最低限度，更快地整合现实<sup>[20]</sup>。减少成本和增加重用。通过以松散耦合的方式公开业务服务，企业可以根据业务要求更轻松地使用和组合服务，可以更轻松地创建、修改和管理它来满足不同时期的需要。

从而应用 SOA 模块化的原理去管理软件的开发流程，使所开发的软件遵循 SOA 的设计思想进行开发和设计。保证了服务模块的可分解性，可组合型，可理解性，可连续性以及可保护性。使得能充分利用现有的资产，集成和管理复杂性变得更为容易。将基础设施和实现发生的改变所带来的影响降到最低限度，更快地整合现实，减少成本和增加重用。通过以松散耦合的方式公开业务服务，企业可以根据业务要求更轻松地使用和组合服务，可以更轻松地创建、修改和管理它来满足不同时期的需要。

## 第四章 软件项目管理系统的设计与实现

### 4.1 系统的概述

本系统的目的是建立一套完善的项目开发 workflow 管理系统。通过人与计算机共同工作的自动化协调、控制和通讯,在计算机化的业务过程中,使所有命令的执行都处于受控状态。在项目开发的过程中,项目参与人员的具体工作情况可以通过时序图的方式体现出来,工作人员每天的工作完成情况以及遇到的技术难题解决方法都会以工作日志形式记录下来,既方便个人对工作的总结,又方便管理层的管理。

同时,从项目确立开始的所有工作都会以文档的形式记录并存档,以备查询。总之,这一系统可以有效的帮助管理人员管理整个工作流程,提高软件开发的工作效率。项目开发的过程中,项目参与人员的具体工作情况可以通过时序图的方式体现出来,工作人员每天的工作完成情况以及遇到的技术难题解决方法都会以工作日志形式记录下来,既方便个人对工作的总结,又方便管理层的管理。从项目确立开始的所有工作都会以文档的形式记录并存档,以备查询。

本软件项目管理系统除了要有管理软件开发的基本功能外,还要在软件开发的需求分析阶段对 SOA 的初始和定义进行管理,在软件的概要设计阶段对 SOA 的设计和构造进行管理,在软件的详细设计和编码阶段对 SOA 的部署进行管理,在软件的测试和维护阶段对 SOA 的后续维护进行管理。这样本软件项目管理系统就充分的应用面向服务(SOA)的思想对所开发的软件进行管理,使所管理开发的软件项目充分应用面向服务(SOA)的思想。

### 4.2 环境的搭建

操作系统: Windows Server 2003

硬件配置: Intel(R) Celeron(R) 2.66Hz, 80G 硬盘, 1G 内存

采用 Mysql 作为数据存储数据库;

采用 Joss/Seam/JSF 技术作为系统的解决方案,使系统具有更好的可扩展性;

采用 Eclips 3.3, 或者支持 JDK6.0, TOMCAT4, SERVERLET, JDBC FOR MYSQL & MSSQL 虚拟主机;

### 4.3 系统的功能设计

系统的业务流程大致分为以下几个方面:分别是对所开发软件需求分析、概要设计、详细设计、编码、测试、维护基本步骤进行管理,也要应用面向服务的思想(SOA)

进行软件开发生的管理。具体介绍如下：

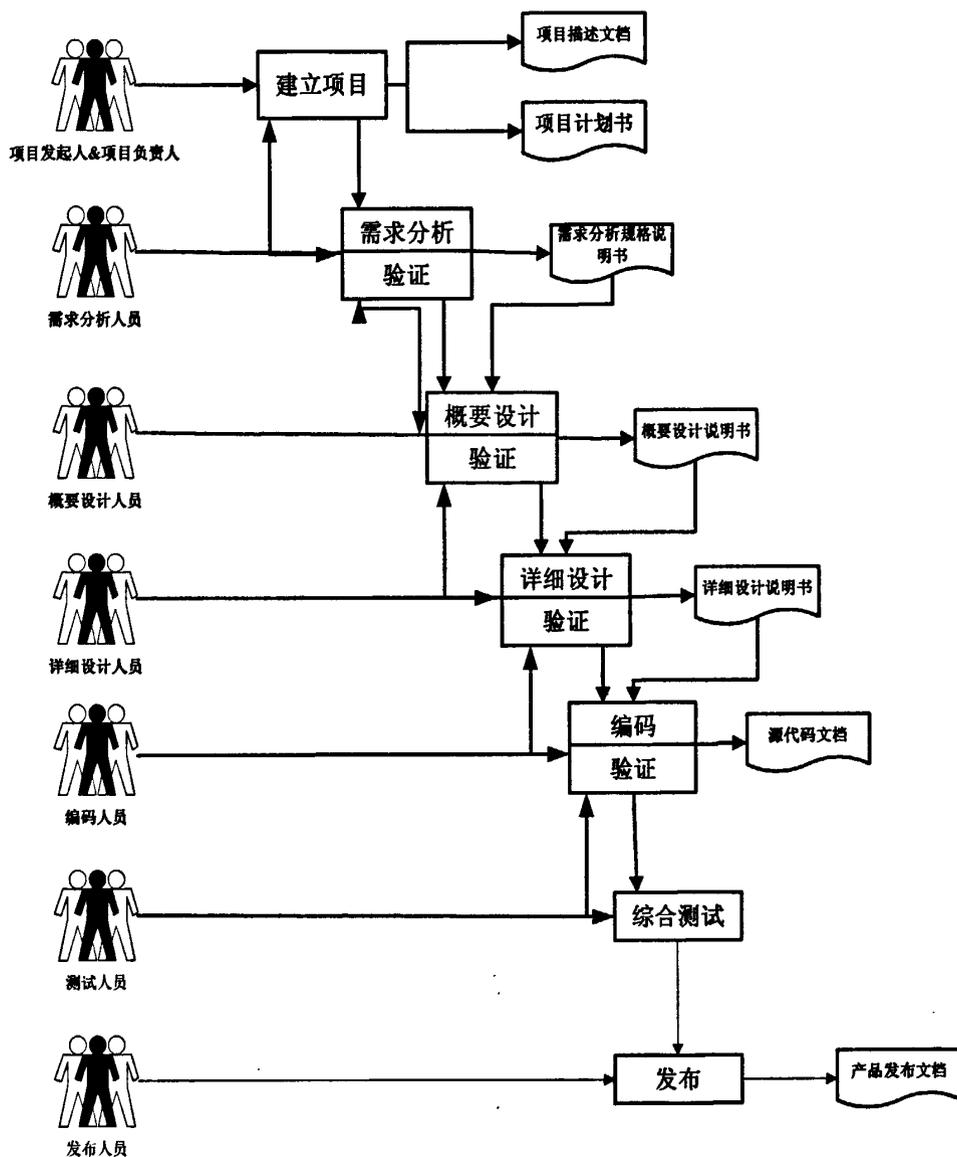


图 4.1 软件项目管理系统整体流程图

### 4.3.1 权限登录

鉴于对系统安全性的要求，用户在登录系统时设有严格的入口控制，判断用户的权限及操作等级，并在进入系统后有严格的访问权限和操作等级控制。

#### 1.入口控制

系统登录设置了用户信息审核机制，用户输入自己的用工 ID、姓名和密码后点击“登录”按钮，此时系统会将接收到的用户信息与系统数据库中的已有用户信息进行核对，如果用户信息和数据库信息是一致的，说明此用户在本系统中存在且用户信息正确，用户即可进入到系统的主页面。如图 4.1 用户登录流程图所示，

#### 2.访问权限控制

系统信息访问权限设置为管理员和普通用户两类。登录系统从用户登录时候开始读取用户的所有信息，直至该用户退出系统。系统预先设定的用户权限将决定系统用户使用本系统的功能范围。如系统管理功能，只有拥有系统管理员权限的用户才能使用，其他用户则无权进行系统管理工作。如图 4.2 用户访问权限控制流程图所示，

### 3.操作等级控制

将工作人员职位设置等级为一般开发人员、项目经理、高层主管三种。登录系统从用户登录时候开始读取用户的所有信息，直至该用户退出系统，用于控制用户对系统某些操作是否合法。如项目人员选取功能只能由项目经理及其以上等级的人可以对项目人员进行更改或者删除操作，普通程序员只能查看信息。

#### 4.3.2 workflow跟踪

workflow跟踪功能使项目管理者能以时序图的形式展示当前正在研发过程中的项目进展情况，也能够对已完成的项目进展情况进行浏览、回顾。

#### 4.3.3 建立项目

这部分主要的功能是显示已确立项目的基本信息，并将需求分析之前相关工作生成的文件进行存档处理，方便用户查询，修改和增加，利于维护相关文档。已立项项目经过本环节的设置才能够进行软件研发工作流的跟踪。

其中主要完成对项目进程的管理即：显示该项目的基本信息（包括项目当前状态）和工作计划（可采用时序图的形式）。状态类型包括：未启动、进行中、暂停、终止，并能够根据软件研发的具体环节具体指明项目在日期及资金上的完成情况。

#### 4.3.4 需求分析管理

这部分主要是为用户提供一个项目开发过程中需求分析部分工作情况的展示平台，用户可以利用该部分的功能维护在整个调研过程中的记录和其他相关文档资料，并对需求分析进行状态监控和管理。如图：



处理：增加新的日志记录，查看已有的调研日志。

输出：需求分析调研日志。

#### 4.3.4.3 需求分析报告维护

说明：本功能主要是对每次调研工作结束后形成的汇总报告（需求分析报告）进行维护。

执行人：需求分析人员

输入：需求分析调研日志和项目相关资料，需求分析报告

处理：增加（修改，存储、提交）汇总报告。

输出：需求分析报告

#### 4.3.4.4 需求协商记录维护

说明：这个功能主要是对与用户协商的记录进行维护

执行人：需求分析人员

输入：需求分析报告

处理：增加（修改，存储、提交）需求协商的记录。

输出：需求协商的记录。

#### 4.3.4.5 需求分析规格说明书的维护

说明：这个功能主要是对需求分析规格说明书进行维护。

执行人：需求分析人员或者项目负责人

输入：审核通过的需求报告

处理：增加、查询。

输出：需求分析规格说明书。

### 4.3.5 概要设计管理

这部分主要是实现对项目开发过程中概要设计部分的管理。实现项目开发概要设计过程中对材料、人员及具体模块设计的规范和管理，并将概要设计所产生的相关文档存档备案，以供后续设计进行查看和修改。它的流程图如下：

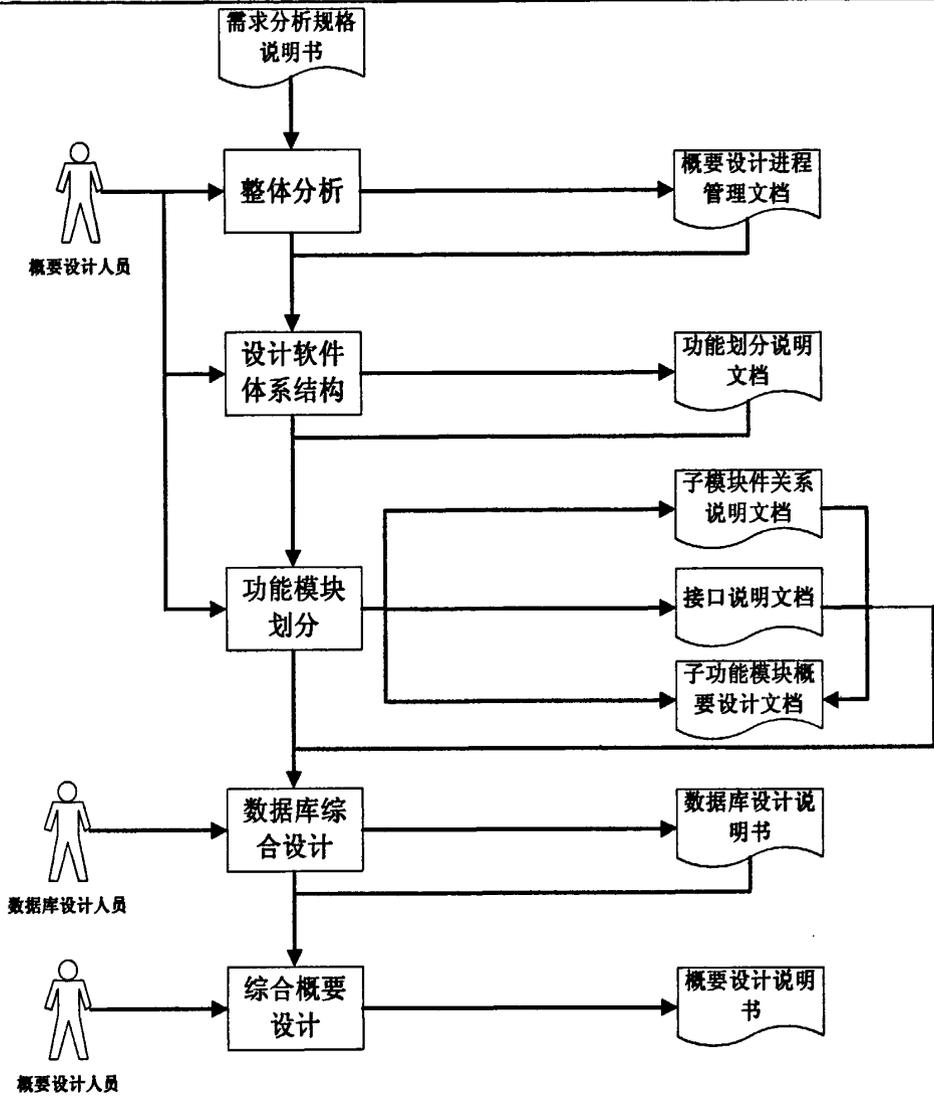


图 4.3 软件项目管理系统概要设计流程图

#### 4.3.5.1 概要设计进程管理

说明：按照需求设计说明书，对项目的概要设计进行整体分析，形成概要设计计划文档,并进行维护；同时实现实际工作时序图，记录概要设计人员任务实际完成情况。

执行人：概要设计人员

输入：需求分析规格说明书

处理：根据需求说明文档的要求，概要设计人员可以对现有的概要设计计划书进行修改、查看、提交等操作。若没有则可增加新的计划书，并对其进行其它操作。（以时序图的形式处理实际工作和计划书，形成实际工作报告和计划书）

输出：.概要设计计划书，概要设计工作时序图

#### 4.3.5.2 软件体系结构划分

说明：这部分是实现对整个软件功能的总体划分。

执行人：概要设计人员

输入：概要设计计划书

处理：依据需求说明文档，划分出项目的各个功能。

输出：功能划分说明文档

#### 4.3.5.3 子模块间关系说明文档维护

说明：对已划分的模块间的关系进行解释说明。

执行人：概要设计人员

输入：功能划分说明文档

处理：将划分的模块间关系说明文档在材料管理功能中存档管理，用于后续设计工作查询、使用。

输出：子模块间关系说明文档

#### 4.3.5.4 接口说明文档维护

说明：对本级模块的接口进行详细描述。

执行人：概要设计人员

输入：功能模块说明文档

处理：将生成的模块间接口说明文档在材料管理功能中存档管理，用于后续设计工作查询、使用。

输出：接口说明文档

#### 4.3.5.5 子功能模块概要设计文档维护

说明：对划分的每一个模块功能进行详细描述。

执行人：概要设计人员

输入：功能划分说明文档、模块间关系说明文档

处理：将生成的功能模块说明文档在材料管理功能中存档管理，用于后续设计工作查询、使用。

输出：子功能模块概要设计文档

#### 4.3.5.6 数据库设计说明书维护

说明：这一部分是对数据库的综合设计，将各个部分的设计汇总。

执行人：数据库设计人员

输入：功能划分说明文档、功能模块说明文档

处理：数据库设计人员根据功能对数据库的需要，对数据库进行总体综合设计。

输出：数据库设计说明书

#### 4.3.5.7 概要设计说明书

说明：概要设计完成后生成的说明文档。

执行人：概要设计人员

输入：功能设计说明书、数据库设计说明书

处理：对整个概要设计完成后，生成概要设计说明书，在材料管理功能中对其进行存档管理，用于后续设计工作的查询、使用。

输出：概要设计说明书、用户手册

### 4.3.6 编码管理

编码主要是完成对软件工程中的编码设计部分的管理。按照编码规范说明文档以及详细设计说明书的具体要求对软件工程中编码设计过程中的环境搭建，任务分配，编码过程，及相关文档存储，编写，存储，查阅等功能进行完善的系统的管理，让编码设计流程更加规范化。

以下是编码设计流程图：

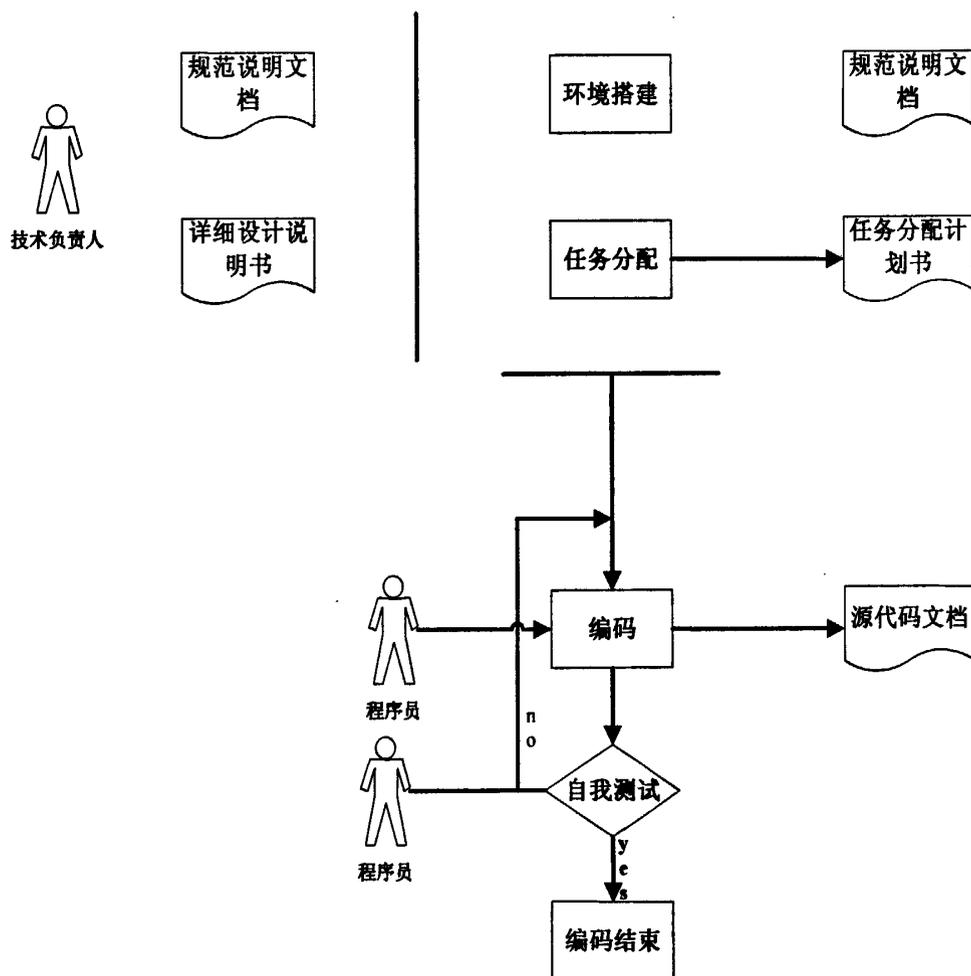


图 4.4 软件项目管理系统编码流程图

## 功能设计

### 4.3.6.1 环境搭建说明书的维护

说明：按照编码规范说明文档以及详细设计说明书的具体要求，进行编码环境的搭

建，并形成环境搭建说明书，并对此进行维护。

执行人：技术负责人

输入：按照编码规范说明文档以及详细设计说明书的具体要求

处理：对环境搭建说明书进行存储，增，删，改，查询等操作。

输出：环境搭建说明书

#### 4.3.6.2 任务分配计划书的维护

说明：按照详细设计说明书的功能模块划分要求，进行编码过程中的人员分配，并加入时序图进行人员任务的显示，形成任务分配计划书并对此进行维护。

执行人：技术负责人

输入：详细设计说明书的功能模块划分要求

处理：对任务分配计划书进行存储，增，删，改，查询等操作。

输出：任务分配计划书，实际工作的时序图

#### 4.3.6.3 源代码文档的维护

说明：对编码阶段的源代码进行存档，形成源代码文档，并对其进行维护。

执行人：程序员

输入：程序员编码

处理：对源代码文档进行存储，增，删，改，查询等操作。

输出：源代码文档

#### 4.3.6.4 编码阶段的自我测试

说明：对编码阶段形成的编码的正确性进行自我测试，改正 BUG，纠正错误，完善编码，使编码符合各部分的要求，并形成新的源代码文档。

执行人：程序员

输入：程序员编码自我测试

处理：对编码阶段形成的编码进行自我测试，进行修改完善，存储形成新的源代码文档。

输出：改正完善后的源代码文档

### 4.4 软件项目管理系统主要功能模块的划分与实现

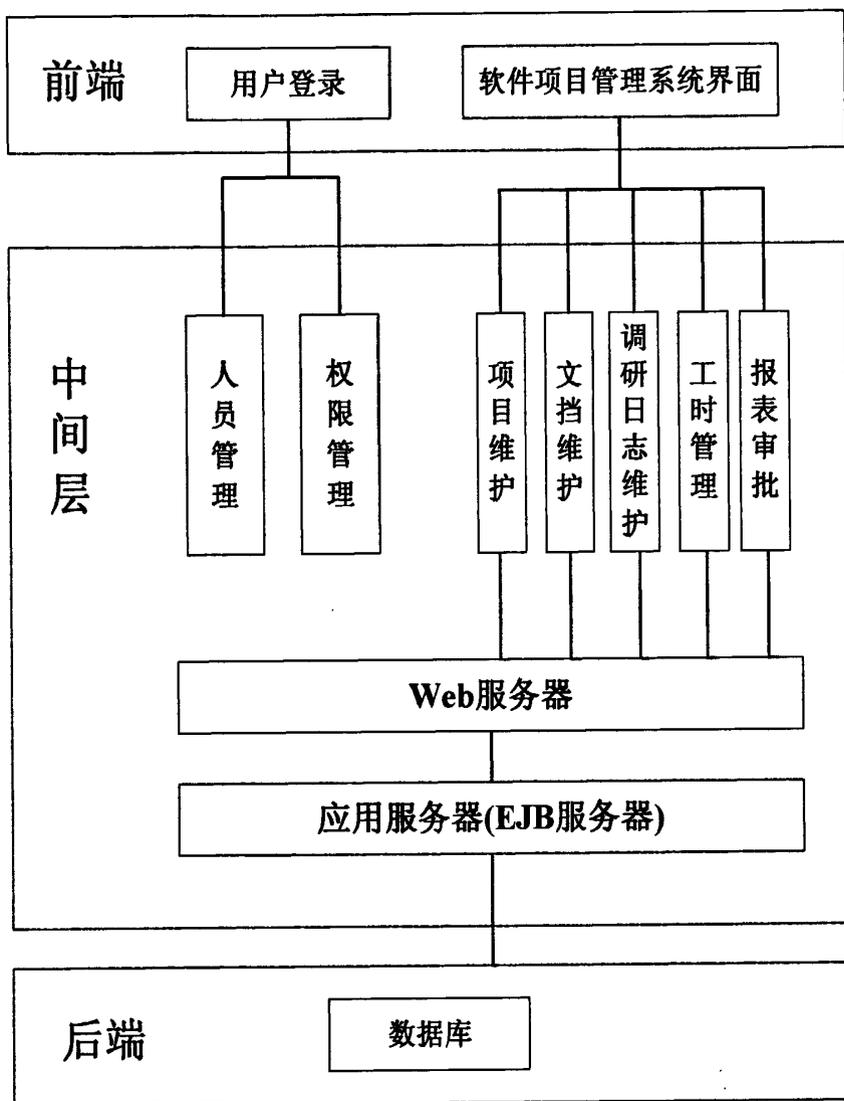


图 4.5 软件项目管理系统主要功能模块图

在完成业务流程的设计后，需要对业务流程进行归纳和分析，并依照业务流程进行功能模块的划分。对本软件项目管理系统主要划分为人员管理、客户管理、开发日志维护、开发文档维护等功能模块。

如图 4.5 所示，本软件项目管理系统的前端需要友好的用户登录界面，以及软件项目管理系统的功能控制界面。系统的主要功能包括：人员管理、人员的权限管理、项目维护、文档维护、调研日志维护、工时管理、客户管理等，这些主要功能都需要在系统的后台操作并管理完成，后端则需要 **Mysql** 提供总体数据库的支持。主要模块的实现简要介绍如下：

#### 4.4.1 登录及权限管理

本软件项目管理系统的用户进行权限登录，系统用户除了要进行用户名、密码的输入外，系统会根据预设的用户权限，系进行功能操作权限的发放。具体操

作用户及功能操作权限如下表所示：

功能权限 用户	日志维护	文档编写	工时管理	项目维护	客户管理	权限管理	数据库维护
程序员	√	√			√		
管理者	√	√	√	√	√		
系统维护人员	√	√	√	√	√	√	√

表 4.1 软件项目管理系统基本权限图

对于权限管理功能则可以在角色设置里增加权限内容，设置新的权限，并在权限设置中选择任意人员拥有该权限的使用，具体如下图：

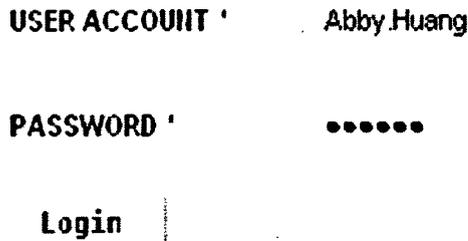


图 4.6 系统登录图

登录及权限管理功能模块主要代码如下：

```

package zof.seam.global;

import org.jboss.seam.Component;

@Name("zof.seam.global.loginValidator")
public class LoginValidator {
    private UserSession userDao = UserSessionBean.instance();
    private Identity identity = Identity.instance();

    public boolean validateLoginAUser() {
        User user = (User) Contexts.getSessionContext().get(SeamPrjConstants.LOGIN_
        if (user != null) {
            return true;
        }
        Credentials credentials = identity.getCredentials();
        String account = credentials.getUsername();
        String password = credentials.getPassword();
        if (account == null || password == null) {
            return false;
        }
        user = userDao.findByAccount(account);
        if (user == null) {
            return false;
        }
        if (password.equals(user.getPassword())) {
            identity.addRole(user.getRole());
            Contexts.getSessionContext().set(SeamPrjConstants.LOGIN_USER_KEY, user)
            return true;
        }
        return false;
    }

    public static LoginValidator instance() {
        LoginValidator instance = (LoginValidator) Component.getInstance("zof.seam.
        return instance;
    }
}

```

#### 4.4.2 人员管理

人员管理功能模块主要是为了对软件公司人员的组织以及人员的配备进行系统化的管理，方便软件公司对公司职员的管理任务的分配等工作。人员管理首先要设置部门名称，可从文档中直接导入部门的相关设置。在设置好部门相关信息后，对相关部门的相关人员进行管理，进行增、删、改、查等操作。

人员管理的主要功能介绍如下：人员基本信息的管理维护，包括新增、删除、修改、查询、列表、浏览。

人员信息新增：新增人员基本信息。

人员信息删除：单个用户删除和批量用户删除。

人员信息修改：修改人员的个人基本信息。

人员信息查询：人员姓名和人员编号作为查询条件进行查询。

人员信息列表：所有客户信息显示出来。

人员信息浏览：选择某一客户浏览客户基本信息。

Resource	Project Role	Effective Start Date	Effective End Date
Amber Ying Zhang	Developer	04/21/2007	03/07/2008
Aviva Yiefang Zhang	QA	04/21/2007	03/07/2008
Bill Zhangchun Zhu	Developer	04/21/2007	03/07/2008
Bob Yongchang Shi	Developer	04/21/2007	03/07/2008
Cindy Yuchen Xu	QA	04/21/2007	03/07/2008
Emily Yiekou Pan	QA	04/21/2007	03/07/2008
Emma Lili Chen	QA	04/21/2007	03/07/2008
Eric Kai Wang	Developer	04/21/2007	03/07/2008
Eve Cayun Xi	QA	04/21/2007	03/07/2008
Helen Yan Wang	Development Lead	04/21/2007	03/07/2008
Jack Chunxia He	Developer	04/21/2007	03/07/2008
Jack Lei Zhu	Developer	04/21/2007	03/07/2008

图 4.7 系统人员设置图

人员管理功能模块主要代码如下：

```

@Stateless
@Name("userSession")
@SuppressWarnings( { "serial" })
public class UserSessionBean extends SeamPrjBaseSessionBean<User> implements UserSession {

    public static UserSession instance() {
        UserSession instance = (UserSession) Component.getInstance("userSession");
        return instance;
    }

    public User findByAccount(String account) {
        String queryName = "zof.seam.userDAO.findByAccount";
        Map<String, Object> params = new HashMap<String, Object>();
        params.put("account", account);
        return getFirstQueryResult(queryName, params);
    }

    public User findById(String id) {
        return entityManager.find(User.class, id);
    }

    public User findByIP(String ip) {
        String queryName = "zof.seam.userDAO.findByIP";
        Map<String, Object> params = new HashMap<String, Object>();
        params.put("ip", ip);
        return getFirstQueryResult(queryName, params);
    }

    public List<User> getAllUserList() {
        String queryName = "zof.seam.userDAO.getAllUserList";
        return getQueryResultList(queryName, (List<Object>) null);
    }

    public List<User> getProjectMemberList(Project project) {
        String queryName = "zof.seam.task.getProjectMemberList";
        Map<String, Object> params = new HashMap<String, Object>();
        params.put("project", project);
        return getQueryResultList(queryName, params);
    }
}

```

#### 4.4.3 报表审批

报表审批功能主要包括对报表上传、查看、及审批等功能。主要完成对员工日常问题进行反馈和管理功能。员工对日常问题、进度用报表的形式向上级进行汇报。上级根据实际情况进行批复。

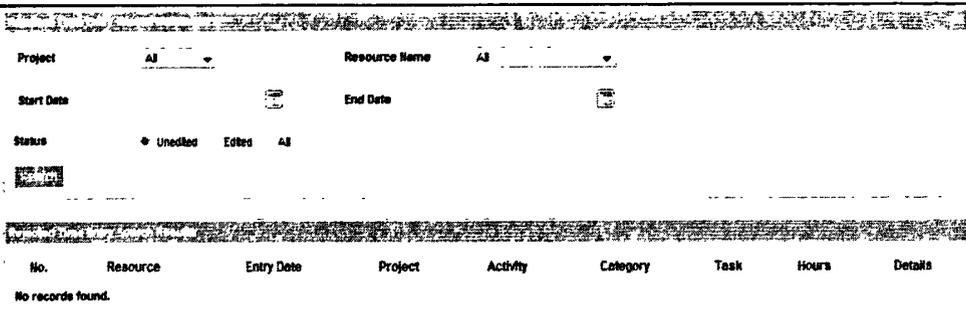


图 4.8 报表审批管理图

报表审批功能模块主要代码如下：

```

public void preUpdateAction() {
    beginConversation();
    if (model.getUser() == null) {
        User loginUser = (User) Contexts.getSessionContext().get(SeamPrjConstants.LOGIN_USER_KEY);
        User user = new User();
        user.setAccount(loginUser.getAccount());
        user.setDisplayName(loginUser.getDisplayName());
        user.setPassword(SeamPrjCoder.instance().decode(loginUser.getPassword()));
        user.setEmail(loginUser.getEmail());
        user.setModel(loginUser.getModel());
        user.setIp(loginUser.getIp());
        user.setIpFrHr(loginUser.getIpFrHr());
        user.setIpToHr(loginUser.getIpToHr());
        model.setUser(user);
    }
}

public void update() {
    User loginUser = (User) Contexts.getSessionContext().get(SeamPrjConstants.LOGIN_USER_KEY);
    User user = model.getUser();
    loginUser.setAccount(user.getAccount());
    loginUser.setDisplayName(user.getDisplayName());
    if (model.isChangePassword()) {
        loginUser.setPassword(SeamPrjCoder.instance().encode(user.getPassword()));
    }
    loginUser.setEmail(user.getEmail());
    loginUser.setModel(user.getModel());
    if (SeamPrjConstants.LOGIN_MODEL_IP.equals(user.getModel())) {
        loginUser.setIp(user.getIp());
        loginUser.setIpFrHr(user.getIpFrHr());
        loginUser.setIpToHr(user.getIpToHr());
    }
    loginUser.setModifyBy(loginUser.getAccount());
    loginUser.setModifyDt(SeamPrjDateTool.getNowDate());
    userDao.update(loginUser);
    endConversation();
    setState("success");
}

```

项目管理功能：项目基本信息的管理维护，包括新增、删除、修改、查询、列表、浏览。项目的立项管理包括项目从发起新建、审批、建立项目相关信息到开始执行，将立项所需的相关资料信息全面管理备案，方便日后的跟踪与管理。

项目执行过程的管理包括了项目任务的分解与执行、项目进度的控制与跟踪、与项目相关的人员、设备、文档及沟通全方位的管理。项目的计划任务进行逐层分解后会将相关的工作内容、人力、进度、设备等信息传递给项目成员，项目成员在整个过程中清晰地了解所要完成及所要配合的项目任务和息，并及时进行沟通汇报与反馈，所有项目相关人员在项目过程中清晰了解整个项目状态，使得沟通配合良好、资源配置合理、监控跟踪有效，同时帮助管理者分析项目进度及项目过程中的问题并及时做出调整，保证项目管理最优化。项目维护的主要功能介绍如下：

项目信息新增：新增项目基本信息。

项目信息删除：单个项目删除和批量项目删除。

项目信息修改：修改未完成的项目的基本信息。

项目信息查询：按照起止日期查询、按照负责人查询和按照合作客户查询。

项目信息列表：所有项目信息显示出来。

项目信息浏览：选择某一项目浏览项目信息。

The screenshot shows a web-based project management interface. At the top, there are four search filters: Project Type (set to All), Business Unit (set to All), Application (set to All), and Project Status (set to All). Below the filters is a table with the following columns: Proj ID, Name, Type, Actual Start Date, Actual End Date, Budget, and Status. The table contains 10 rows of data. At the bottom left, it says '37 record(s) found, displaying 10 record(s), from 1 to 10.' At the bottom right, there is a navigation bar with 'Page 1 / 4'.

Proj ID	Name	Type	Actual Start Date	Actual End Date	Budget	Status
5255	GEN	Internal	02/12/2007		0.0	Active
5257	GCI-CD1	External	04/21/2007	03/07/2008	1217.0	Closed
5258	GCI-CD2	External	04/21/2007	03/07/2008	2011.0	Closed
10621	LEAVE	Internal	04/23/2007		0.0	Active
10218	BAS-321	Internal	06/04/2007		0.0	On-Hold
5251	VOD-R3	External	06/25/2007	07/02/2007	550.0	Cancelled
5251	SHY-RE	External	07/23/2007		313.0	Active
5255	SHY-R3	External	07/23/2007	07/11/2008	465.0	Closed
10595	HL-R2	External	09/01/2007	12/13/2007	310.0	Closed
5254	GAI-R8	External	09/02/2007	01/15/2008	240.0	Closed

图 4.9 项目查询图

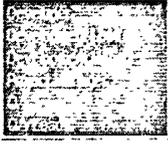
Category Code *	DBA/SERVER/BULD/SOF	Priority *	50
Status *	<input checked="" type="radio"/> Active <input type="radio"/> Inactive	Project Roles *	
Description *	<input type="text" value="DBA/Server/Buid/Software"/>		
Comments	<input type="text"/>		



图 4.10 系统项目维护图

项目维护功能模块主要代码如下：

```

@Name("projectAction")
@Scope(ScopeType.CONVERSATION)
@SuppressWarnings({ "serial" })
public class ProjectAction extends SeamPrjBaseAction {

    private ProjectSession prjSession = ProjectSessionBean.instance();
    private UserSession userSession = UserSessionBean.instance();
    private ProjectModel model = ProjectModel.instance();

    @RequestParameter
    private String pPrjId;

    @Override
    public void before() {
        ProjectModel.State state = (ProjectModel.State) Contexts.getConversationContext().get(ProjectModel.State.KEY);

        if (state == null) {
            state = new ProjectModel.State();

            model.getProjects().clear();
            List<Project> projects = prjSession.getAllProjects(model.getLoginuser());
            model.getProjects().addAll(projects);

            model.getMembersItems().clear();
            List<User> users = userSession.getAllUserList();
            for (User user : users) {
                SelectItem item = new SelectItem(user, user.getDisplayName());
                model.getMembersItems().add(item);
            }
        }

        if (model.getProjects().size() > 0) {
            if (model.getProject() == null && pPrjId != null) {
                for (Project p : model.getProjects()) {
                    if (pPrjId.equals(p.getId())) {
                        p.setSelected(true);
                        model.setProject(p);
                        break;
                    }
                }
            }
            if (model.getProject() == null) {
                Project fp = model.getProjects().get(0);
                fp.setSelected(true);
                model.setProject(fp);
            }
        } else {
            model.setProject(null);
        }

        if (model.getNewProject() == null) {
            model.setNewProject(new Project());
        }
        Contexts.getConversationContext().set(ProjectModel.State.KEY, state);
    }
}

```

#### 4.4.5 调研日志维护

调研日志维护功能模块是为软件公司程序员的日常开发所设计。主要功能是对程序员日常日志的记录，问题的反馈进行管理，这样可以使软件的开发更加稳定。调研日志维护的主要功能介绍如下：

日志新增：新增一条日志记录。

日志删除：单个日志删除和批量日志删除。

日志信息修改：修改未完成的日志的基本信息。

日志信息查询：按照日期查询、按日志内容查询、按项目相关查询和按文档代号查询。

日志信息列表：所有当前项目的调研日志列表显示在本页面。

日志信息浏览：选择某一项目浏览项目信息。

No.	Resource	Entry Date	Project	Activity	Category	Task	Hours	Details
1	Tom Yanbin Fan	07/19/2008	GEN	208759	TRAINING	TRAINING - SELF-STUDY	7.0	Did fn229511 activity
2	Tom Yanbin Fan	07/17/2008	GEN	208759	TRAINING	TRAINING - SELF-STUDY	7.0	Did fn229511 activity
3	Tom Yanbin Fan	07/17/2008	GEN	206281	ENGLISH CLASS	ENGLISH CLASS	1.0	ENGLISH CLASS
4	Tom Yanbin Fan	07/16/2008	GEN	208759	TRAINING	TRAINING - SELF-STUDY	8.0	Did fn229511 activity
5	Tom Yanbin Fan	07/15/2008	GEN	208759	TRAINING	TRAINING - SELF-STUDY	5.0	Did fn229510 activity
6	Tom Yanbin Fan	07/15/2008	GEN	208759	TRAINING	TRAINING - ATTENDEE	2.0	training JIRA
7	Tom Yanbin Fan	07/15/2008	GEN	206281	ENGLISH CLASS	ENGLISH CLASS	1.0	ENGLISH CLASS
8	Tom Yanbin Fan	07/14/2008	GEN	208759	TRAINING	TRAINING - SELF-STUDY	8.0	Did fn229510 activity
9	Tom Yanbin Fan	07/11/2008	GEN	208759	TRAINING	TRAINING - ATTENDEE	2.5	training J2ME technology
10	Tom Yanbin Fan	07/11/2008	GEN	208759	TRAINING	TRAINING - SELF-STUDY	5.0	Did fn229510 activity

160 record(s) found. displaying 10 record(s), from 1 to 10.

图 4.11 调研日志维护图

#### 4.4.6 文档维护

文档维护功能模块主要功能是对软件项目开发过程中主要的项目文档进行管理，如需求分析说明书、概要设计说明书等。方便对项目的主要文档进行管理，对项目的规格化开发起着重要的作用。

文档管理功能介绍如下：文档基本信息的管理维护，包括新增、删除、修改、查询、列表、浏览、调研文档相关日志的选择。

调研文档新增：新增调研文档基本信息。

调研文档删除：单个调研文档删除和批量用户删除。

调研文档修改：修改已知单调研文档基本信息。

调研文档查询：文档名字作为查询条件进行查询。

调研文档列表：所有调研文档显示出来。

调研文档浏览：选择某一调研文档浏览调研文档基本信息。

调研文档相关日志的选择：选择与调研文档相关的

Resource: Jack Churoua He								
No.	Entry Date	Project	Activity	Category	Task	Hours	Details	Reject Reason
1	07/21/2008	BEC-R1	230155	DEVELOPMENT	DEVELOP - INITIAL	7.00	develop this activity	Day total < 8 hrs
2	07/22/2008	BEC-R1	230155	DEVELOPMENT	DEVELOP - INITIAL	8.00	develop this activity	
3	07/22/2008	GEN	208759	TRAINING	TRAINING - ATTENDEE	2.00	EXT training	Wrong Category/Act
4	07/23/2008	BEC-R1	230155	DEVELOPMENT	DEVELOP - INITIAL	6.00	develop this activity	

have not 40 Hours this week

Time Entry  
Comments

Total Hours: 23.00

图 4.12 文档维护图

文档维护功能模块主要代码如下：

```

import java.util.ArrayList;

@Name("taskModel")
@Scope(ScopeType.CONVERSATION)
@SuppressWarnings({ "serial" })
public class TaskModel extends SeamPrjBaseModel {

    public static TaskModel instance() {
        TaskModel instance = (TaskModel) Component.getInstance("taskModel");
        return instance;
    }

    public static class State implements IState<TaskModel> {
        public static final String KEY = "taskModel.State";
        public String projectId;
        public String milestoneId;
    }

    private final List<SelectItem> projectItems = new ArrayList<SelectItem>();
    private Project project;
    private final List<SelectItem> milestoneItems = new ArrayList<SelectItem>();
    private Milestone milestone;
    private final List<Task> taskList = new ArrayList<Task>();
    private Task task;
    private Task newTask;

    private final List<SelectItem> dependItems = new ArrayList<SelectItem>();
    private final List<SelectItem> prjMembers=new ArrayList<SelectItem>();

    public List<SelectItem> getPrjMembers() {
        return prjMembers;
    }

    public List<SelectItem> getDependItems() {
        return dependItems;
    }

    public Project getProject() {
        return project;
    }
}

```

#### 4.4.7 工时管理

工时管理模块对软件开发进度的监督和规划都起着重要的作用。项目的管理人员可以利用甘特图来浏览并了解指定项目的开发进度,并可了解任务的情况以及设置里程碑等。

可以对项目进行多级分解为多个子任务,并对整体任务进度和子任务定义责任人及相关资源、信息,跟踪项目进度和完成情况,从不同角度监控项目,以便及时对项目做出调整。

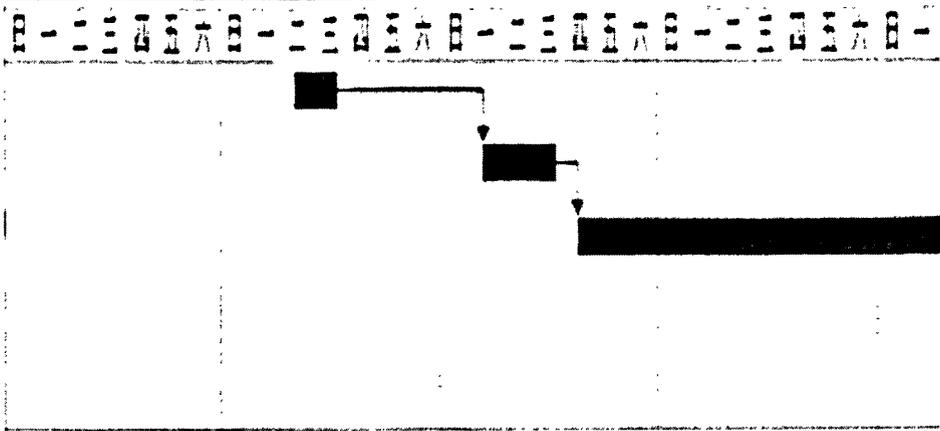


图 4.13 工时管理图

工时管理功能模块主要代码如下：

```

@Name("milestoneAction")
@Scope(ScopeType.CONVERSATION)
@SuppressWarnings({ "serial" })
public class MilestoneAction extends SeamPrjBaseAction {

    private ProjectSession prjSession = ProjectSessionBean.instance();
    private MilestoneSession mlsSession = MilestoneSessionBean.instance();
    private MilestoneModel model = MilestoneModel.instance();

    @RequestParameter
    private String pPrjId;
    @RequestParameter
    private String pMlsId;

    @Override
    public void before() {
        MilestoneModel.State state = (MilestoneModel.State) Contexts.getConversationContext().get(MilestoneModel.State.KEY);
        if (state == null) {
            state = new MilestoneModel.State();

            model.getProjectItems().clear();
            List<Project> prjList = prjSession.getAllProjects(model.getLoginuser());
            for (Project p : prjList) {
                SelectItem item = new SelectItem(p, p.getPrjCode() + " : " + p.getPrjName());
                model.getProjectItems().add(item);
            }
        }
        if (model.getProjectItems().size() > 0) {
            if (model.getProject() == null && pPrjId != null) {
                for (SelectItem item : model.getProjectItems()) {
                    Project p = (Project) item.getValue();
                    if (pPrjId.equals(p.getId())) {
                        model.setProject(p);
                        break;
                    }
                }
            }
        }
    }
}

```

```

    if (model.getProject() == null) {
        SelectItem first = model.getProjectItems().get(0);
        Project fp = (Project) first.getValue();
        model.setProject(fp);
    }
} else {
    model.getMilestoneList().clear();
}

Project curPrj = model.getProject();
if (curPrj != null) {
    String prePrjId = state.projectId;
    if (prePrjId == null || !prePrjId.equals(curPrj.getId())) {
        model.getMilestoneList().clear();
        List<Milestone> mlsList = mlsSession.getProjectMilestoneList(curPrj);
        model.getMilestoneList().addAll(mlsList);
    }
} else {
    model.getMilestoneList().clear();
}

if (model.getMilestoneList().size() > 0) {
    if (model.getMilestone() == null && pMlsId != null) {
        for (Milestone m : model.getMilestoneList()) {
            if (pMlsId.equals(m.getId())) {
                m.setSelected(true);
                model.setMilestone(m);
                break;
            }
        }
    }
} else {
    if (model.getMilestone() == null) {
        Milestone fm = model.getMilestoneList().get(0);
        fm.setSelected(true);
        model.setMilestone(fm);
    }
} else {
    model.setMilestone(null);
}

if (curPrj != null) {
    if (model.getNewMilestone() == null) {
        model.setNewMilestone(new Milestone());
    }
} else {
    model.setNewMilestone(null);
}

state.projectId = (curPrj == null ? null : curPrj.getId());
Contexts.getConversationContext().set(MilestoneModel.State.KEY, state);
}

```

#### 4.5 系统数据库的设计

### 4.5.1 数据库语言的选取

本系统主要是利用数据库设计工具 **Mysql** 进行数据库设计，构造数据库概念模型并生成其相应 **SQL** 代码，在 **Mysql** 数据库中导入 **SQL** 代码执行可得到所有数据库表。之所以应用 **Mysql** 作为数据库是由于其所占用的资源较少且速度快并总体拥有成本低，尤其是源码较为开放这一特点是选择它作为本系统数据库设计工具的主要原因。

### 4.5.2 数据库的连接

在 **Seam** 框架中，与数据库的接口工作是由.xml文件完成的。需要编写 **Seam** 框架中 **mysql-ds.xml** 并放在 **server/default/deploy** 文件夹下与 **Mysql** 数据库进行连接。具体如下：

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE datasources
    PUBLIC "-//JBoss//DTD JBoss JCA Config 1.5//EN"
    "http://www.jboss.org/j2ee/dtd/jboss-ds_1_5.dtd">

<datasources>

    <local-tx-datasource>
        <jndi-name>myds</jndi-name>
        <connection-url>jdbc:mysql://localhost:3306/seam?useUnicode=true#38;amp;characterEncoding=UTF-8</connection-url>
        <driver-class>com.mysql.jdbc.Driver</driver-class>
        <user-name>root</user-name>
        <password>mysql</password>

        <exception-sorter-class-name>
            org.jboss.resource.adapter.jdbc.vendor.MySQLExceptionSorter
        </exception-sorter-class-name>
        <metadata>
            <type-mapping>mysql</type-mapping>
        </metadata>

    </local-tx-datasource>

</datasources>
```

### 4.5.3 数据库表的设计

人员管理分为两个部分，人员列表和新增人员。人员列表从 **staff** 表中读出人员的基本信息并显示出来，对人员信息进行修改和删除后将内容重新写入 **staff** 表中；在新增人员中填写人员基本信息，并写入 **staff** 表中。

**staff** 表 (人员表)

东北师范大学硕士学位论文

字段名	类型	长度	是否为空	说 明	主键/ 外键	备注
staff_GUID	varchar	36	否	人员 GUID	主键	
staff_Code	varchar	128	否	人员前台标号		
staff_Name	varchar	256	否	人员姓名		
staff_Sex	int		否	人员性别(数据字典)		
staff_Brithday	datetime		可以为空	人员生日		
staff_Address	varchar	1024	可以为空	人员住址		
staff_Educational	varchar	2048	可以为空	人员履历(数据字典)		
staff_Politics	varchar	128	可以为空	人员政治面貌(数据字典)		
staff_Tel	varchar	128	否	人员电话(移动)		
staff_Tel	varchar	128	否	人员电话(固定)		
	varchar	4	可以为空	内线电话		
staff_Position	varchar	512	否	人员职务(数据字典)		
staff_Department	varchar	512	否	人员隶属部门(数据字典)		
	varchar	1024	可以为空	其他备注信息		

表 4.2 人员表

客户管理分为客户信息列表和新增客户两个部分，在客户信息列表中，从 client 表中读出客户的基本信息，并在页面显示出来，点击新增将会跳转到客户新增也面，在此可以添加客户，并将其写如 client 表中。

client 表 (客户表)

字段表	类型	长度	是否为空	说 明	主键/ 外键	备注
client_GUID	varchar	36	否	客户 GUID	主键	
client_Name	varchar	256	否	客户姓名		
client_Sex	varchar	20	否	客户性别(数据字典)		001001
client_Company	varchar	512	否	客户所在公司		
client_Tel	varchar	128	可以为空	客户电话		与 mail 不能同时为空
client_Email	varchar	512	可以为空	客户 E-mail		
client_Homepage	varchar	512	可以为空	客户主页		
			可以为空	其他备注信息		

表 4.3 客户表

将调用人员列表(staff 表)中“员工的 GUID”(staff\_GUID)、客户信息列表(client 表)中的“客户 GUID”(client\_GUID)和附件列表(attachment 表)中的“附件 GUID”(attachment\_GUID)。本页面将产生的数据信息如下表, 这些信息将存入 project 表中。  
project 表(项目表)

字段表	类型	长度	是否为空	说明	主键/外键	备注
project_ID	int	36	否	项目 ID	主键	
project_Uicode	varchar	128	否	项目前台代号		
project_Name	varchar	256	否	项目名称		
project_Description	varchar	2000	可以为空	项目描述		
project_Date	datetime		否	项目建立日期		自动获取
staff_GUID	char	36	否	员工 GUID	外键	
client_GUID	char	36	否	客户 GUID	外键	
attachment_GUID	char	36	否	附件 GUID	外键	
				备注信息		

表 4.4 项目表

将调用项目表(project 表)中的项目 ID(project\_ID)、人员列表(staff 表)中“员工的 GUID”(staff\_GUID)、客户信息列表(client 表)中的“客户 GUID”(client\_GUID), 本页面将产生的数据如下表, 这些数据将存入 diary 表中。

字段表	类型	长度	是否为空	说明	主键/外键	备注
diary_GUID	varchar	36	否	日志 GUID	主键	
diary_Uicode	varchar	128	否	日志前台代号		
diary_Name	varchar	256	否	日志名称		
diary_Content	varchar	4000	否	日志内容		
diary_Date	datetime		否	日志日期		
diary_Problem	varchar	2000	可以为空	待解决问题		
client_GUID	varchar	32	否	客户 GUID	外键	
staff_GUID	varchar	32	否	人员 GUID	外键	
project_ID	varchar	32	否	项目 ID	外键	
				其他备注信息		

表 4.5 日志维护表

将调用项目表(project 表)中的项目 ID(project\_ID)、人员列表(staff 表)中“员工的 GUID”(staff\_GUID)、客户信息列表(client 表)中的“客户 GUID”

(client\_GUID), 日志表中的“日志列表”(diary\_GUIDList) 本页面将产生的数据如下表, 这些数据将存入 file 表中。

file 表 (文档表)

字段名	类型	长度	是否为空	说明	主键/外键	备注
file_GUID	varchar	32	否	文档 GUID	主键	
file_Code	varchar	128	否	文档前台号		
file_Name	varchar	256	否	文档名		
file_Content	varchar	4000	否	文档内容		
file_Date	datetime		否	文档编写时间		自动获取
client_GUID	varchar	32	否	客户 GUID	外键	
diary_GUIDList	varchar	128	否	日志列表	外键	
project_ID	varchar	32	否	项目 ID	外键	
staff_GUID	varchar	32	否	人员 GUID	外键	

表 4.6 文档维护表

## 第五章 总结与展望

### 5.1 总结

本文细致研究了面向服务（SOA）的思想、软件工程的流程、Jboss Seam 等开发工具。再对此进行了细致研究、探讨、学习基础上调查并分析了多个小型软件公司的软件项目管理系统，总结了多个软件项目管理系统的在软件项目管理流程上的优点及缺点。在此基础上，运用 Jboss Seam 等开发工具、按照软件工程开发软件系统的流程、并融入 SOA 的思想去管理软件的开发、集合了多数软件管理系统的优点，进行了本软件项目管理系统的初步开发。

本系统的目的是建立一套完善的项目开发 workflow 管理系统。通过人与计算机共同工作的自动化协调、控制和通讯，在计算机化的业务过程上，使所有命令的执行都处于受控状态。在项目开发的过程中，项目参与人员的具体工作情况可以通过时序图的方式体现出来，工作人员每天的工作完成情况以及遇到的技术难题解决方法都会以工作日志形式记录下来，既方便个人对工作的总结，又方便管理层的管理。

同时，从项目确立开始的所有工作都会以文档的形式记录并存档，以备查询。总之，这一系统可以有效的帮助管理人员管理整个工作流程，提高软件开发的工作效率。

项目开发的过程中，项目参与人员的具体工作情况可以通过时序图的方式体现出来，工作人员每天的工作完成情况以及遇到的技术难题解决方法都会以工作日志形式记录下来，既方便个人对工作的总结，又方便管理层的管理。从项目确立开始的所有工作都会以文档的形式记录并存档，以备查询。

本系统一个创新点就是将 SOA 这种思想应用到软件项目开发管理的体系中。简单的说通过 SOA 思想在本系统中的应用，使在本系统管理下的软件开发流程遵循 SOA 的规则，增强的所管理开发软件的面向服务的特性，增加代码的重用性，使所开发的项目可以跨平台，跨语言的结合。大大缩短的软件开发的工期，加强了软件开发面向服务的逻辑性。

### 5.2 展望

IT 行业是一个飞速发展的行业，随之，软件项目管理系统也跟着迅速的发展和进步着。本软件项目管理系统在我们研究小组的不断的探索和学习开发中虽然有了一定的成果。但还有很多地方需要进一步的改善和完善。

1. 由于时间、技术、人员的限制，本系统只完成的初步的开发，还有跟多功

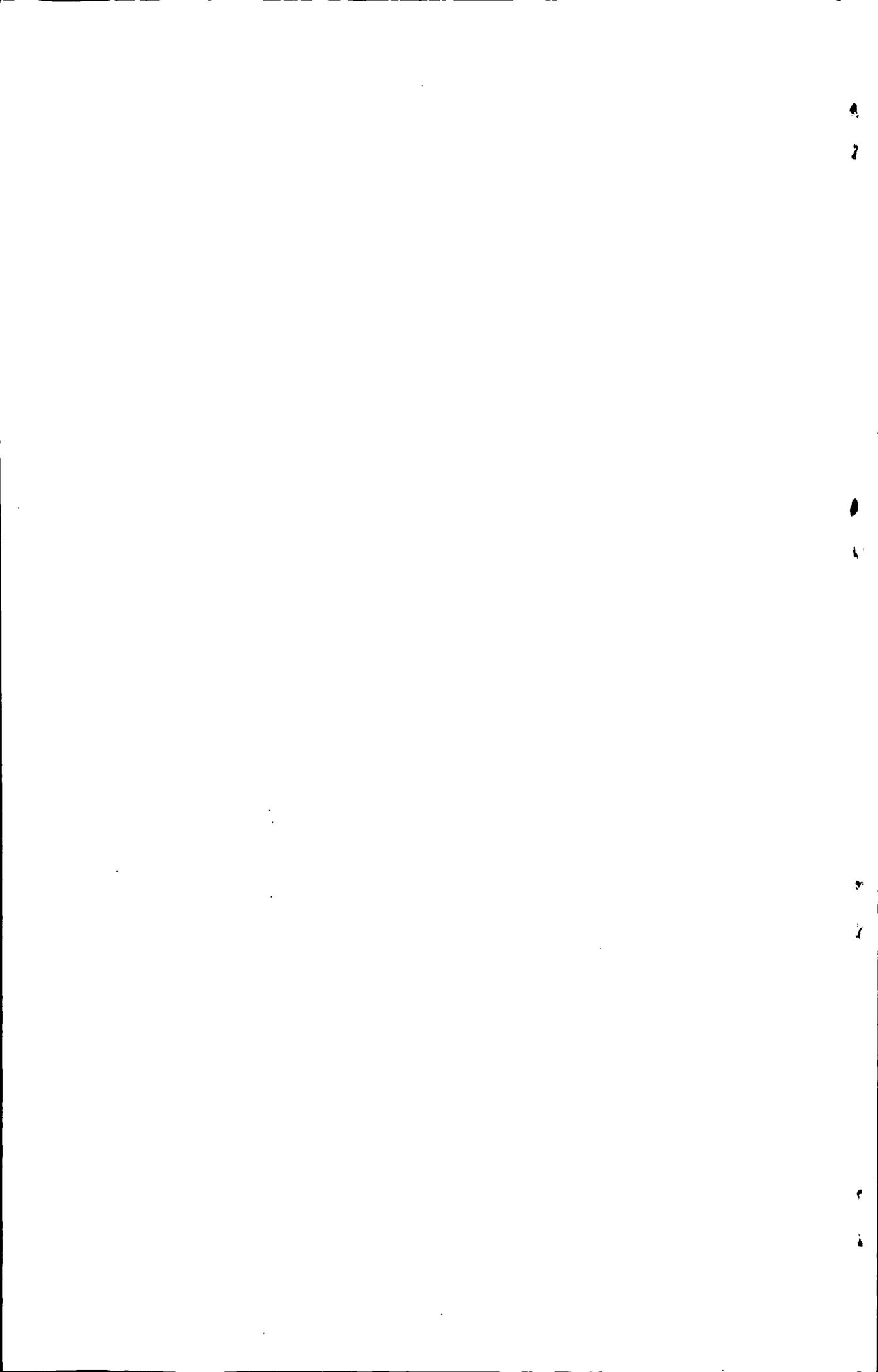
能需要完善和设计。

2.本文关于如何运用 SOA 的思想去管理软件项目的开发,在理论上还研究的不够成熟,还需要进一步研究和完善,并逐渐接受实践的检验。

3.对于 Jboss Seam 开发框架,JSF workflow跟踪还需要进一步学习和运用。希望在今后的其他软件开发工作中,能完全灵活运用这些开发工具去参与软件项目的快速开发与设计。

## 参考文献

- [1] 张巍.《面向工程现场信息的采集装置设计及应用系统开发》.重庆大学硕士学位论文,2008.
- [2] 石永东, 胡树华.《我国工程管理软件的现状及发展对策》.建筑管理现代化,2002.
- [3] Hoyle, C.R., Anthropogenic influence on SOA and the resulting radiative forcing. In Proc. of DaWaK, 2008.
- [4]刘欣怡.《软件工程》.清华大学出版社,2007.
- [5]罗先文等.《软件工程实务》.重庆大学出版社,2005.
- [6] Roger S. Pressman, Software Engineering: A Practitioner's Approach ( Fourth Edition), McGraw-Hill, 1997
- [7] Mary Shaw, David Garlan, Software Architecture, Prentice-Hall, 1996
- [8] Geoffery James, 编程之道 (郭海等译), 清华大学出版社, 2009
- [9] Friesen,A,ASMs in service oriented architectures , 2008.
- [10] Bychkov, I.V.. A service-oriented architecture for some problems.2007
- [11]陈根洪.《基于 SOA 的企业物资资源管理系统设计与开发》.华东师范大学硕士学位论文,2009.
- [12]仲萃豪.《SOA 的十大技术理论体系》.中国计算机用户,2006.
- [13] SOA 的生命周期,<http://se.csai.cn>.
- [14] Chatterjee, A. e-Data Testing service-oriented architectures, 2008.
- [15] 张洪斌.《JBoss 平台上的 Java EE 程序开发指南》.清华大学出版社,2007.
- [16] 李宜南.《浅论软件项目管理》.教育经济研究,2008.
- [17] 邓杰超.《软件项目管理探析》.广州土地管理房产学校,2006.
- [18]成功规划 SOA: 进行长期 SOA 规划, <http://se.csai.cn>.
- [19] Liqing, F., Senthil Kumar, A., Integrated fixture design and analysis system based on service-oriented architecture , 2008.
- [20] Eric Newcomer .《Understanding SOA with Web Services 中文版》,电子工业出版社,2006.
- [21] SOA 模块化的概念,<http://se.csai.cn>.
- [22]潘杰,陈德,周咨聪.《基于 SOA 开发框架的异构 workflow 集成方案》.微计算机应用,2008 年 29 卷第 1 期
- [23]傅向华,明仲,彭小刚.《基于 SOA 的流程与数据关联模型研究》.计算机应用研究,2008 年 25 卷第 1 期.
- [24]杨文龙, 姚淑珍, 吴云, 软件工程, 电子工业出版社, 1997
- [25]使用可重用资产构建 SOA 应用程序. 摘自 IBM Developerworks.
- [26]李宜南,潘培亮.《浅论软件项目管理》.教育经济研究,2008.
- [27]岑磊.《软件项目管理的探究》.Http: [www.hhjy.net](http://www.hhjy.net).
- [28]潘杰,陈德,周咨聪.《基于 SOA 开发框架的异构 workflow 集成方案》.微计算机应用,2008 年 29 卷



## 致 谢

时光飞逝，在东北师范大学理想信息技术研究院的三年研究生时光即将渡过。短短的三年研究生生活使我受益匪浅，在理想信息技术研究院的培养下，我的学术和思想道德水平都得到了长足的进步。

在这里，我最要感谢的是我的导师李献业教授。李老师广博的学识和在学业上广博的知识和严密的逻辑思维，在科研实践中的身体力行、踏踏实实的工作作风都使我受益匪浅。三年来他对我的悉心教导，对我起到了相当大的帮助，李老师的思想品德和工作作风也对我今后的工作和学习指引着方向。

我还要感谢研究院的老师对我的指导和教育，以及数据库实验室老师、同学、同事对我的鼓励和帮助。在论文即将完成之际，我要对他们表示最真诚的谢意！

即将结束再次学习的生活，相信等待我的是一片充满机遇、风险与快乐的土地；也相信我和同学们的事业必将如涅槃之凤、浴火之凰；更加相信，今天的努力会成就我们前程似锦的未来！“风雨不改凌云志，振衣濯足展襟怀。行方智圆煅内蕴，海阔天空铸宏图。”

