

# 软件应用系统设计与实施方案

## 目 录

1	软件系统架构设计	6
1.1	概要说明	6
1.2	系统特点	6
1.2.1	根据优化流程开发	6
1.2.2	充分利用现有资源	6
1.2.3	先进的设计理念	6
1.2.4	开放式的可扩展性	6
1.2.5	与现有系统轻松衔接	7
1.2.6	可信赖的高可靠性	7
1.3	总体体系架构	7
1.3.1	基于组件的 S O 系统应用架构	7
1.3.2	系统技术框架	8
1.3.3	其他重要问题	10
1.4	主平台解决方案	11
1.4.1	基于工作流的业务流程管理	12
1.4.1.1	监控管理	
1.4.1.2	工作项服务	
1.4.1.3	日志服务	
1.4.2	业务规则管理	14
1.4.3	主平台和各子系统的接口	14
1.4.4	多级基于角色的权限管理	14
1.5	数据模型	15
1.5.1	数据建模原则	15
1.5.2	数据建模方法	16
1.5.3	数据质量管理	18
1.5.4	数据存储方式	18
1.5.5	其他重要问题	18
1.6	用户界面	18
1.6.1	用户界面设计原则	18
1.6.2	用户界面层设计技术	19
2	概要设计说明	
2.1	概述	20
2.2	设计原则	20
2.2.1	统一设计原则	20
2.2.2	先进性原则	20
2.2.3	高可靠 高安全性原则	21
2.2.4	标准化原则	21
2.2.5	成熟性原则	21
2.2.6	适用性原则	21
2.2.7	可扩展性原则	21
2.3	系统功能综述	21
2.3.1	主控平台	21
2.3.2	房屋图元信息	21
2.3.3	房屋基础信息	22
2.3.4	楼盘表	22
2.3.5	房屋权属信息	22
2.3.6	房屋地址库信息	22

2.3.7	统计分析	22
2.4	重点子系统解决方案	23
2.4.1	子系统解决方案	23
2.4.1.1	子系统架构图	
2.4.1.2	子系统预受理组件业务流程图	
3	接口、部署及迁移实施方案	
3.1	接口方案	24
3.2	系统部署方案	24
3.3	系统硬件部署方案	25
3.3.1	硬件部署图	25
3.3.2	网络拓扑结构	25
3.3.2.1	数据库层	
3.3.2.2	存储层	
3.3.2.3	应用层	
3.3.2.4	发布层	
3.3.3	内外网交换系统	31
3.3.4	网络安全	33
3.4	系统迁移实施方案	33
3.4.1	数据迁移	34
3.4.1.1	数据迁移需求分析	
3.4.1.2	迁移规则制定	
3.4.1.3	数据资源规划和清理	
3.4.1.4	数据迁移工具的选择	
3.4.1.5	数据迁移试迁及完善	
3.4.1.6	正式迁移	
3.4.2	系统切换及过度时间计划	35
3.4.2.1	风险分析	
3.4.2.2	切换方案	
4	平台技术标准与规范	
5	应用系统培训方案	
5.1	万里红有限公司的培训优势	
5.2	基础条件	
5.3	培训对象及目标	
5.4	管理层培训	38
5.5	系统管理人员培训	38
5.6	普通用户培训	39
5.7	外地代理商培训	39
5.8	约束条件	40
5.9	培训结果的评估	40
5.10	培训方式	40
6	平台的建设建议	
7	所需的第三方产品	
8	项目开发和管理工具	
9	软件生命周期各阶段的工艺、方法	
9.1	项目启动阶段	44
9.2	需求分析阶段	45
9.3	系统设计阶段	46
9.4	系统实现阶段	
9.5	集成测试阶段	49
9.6	系统测试阶段	49

9 . 7 系统交付阶段	51
9 . 8 系统维护阶段	51
1 0 项目实施方法	
1 0 . 迭代式软件开发模式	52
1 0 . 为什么要以迭代方式开发	52
1 0 . 迭代式方法的优点	53
1 1 项目实施各个阶段的进度计划、成果及交付物说明	
7 ) 系统交付阶段	
1 2 项目管理方案	
1 2 . 项目组织机构	61
1 2 . 1 组织结构及组织图	61
1 2 . 1 投入人力的职能及责任限度	61
1 2 . 2 范围控制	62
1 2 . 3 进度控制	63
1 2 . 4 质量保证	64
1 2 . 4 . Q1A经理	64
1 2 . 4 . Q2A工程师	65
1 2 . 5 沟通管理	66
1 2 . 5 项目主管	66
1 2 . 5 项目组	67
1 2 . 5 . QA工程师	67
1 2 . 6 配置管理	69
1 2 . 7 文档范本	69
1 2 . 8 风险控制	
1 2 . 8 项目风险	
在出现不可修复的危害之前准备修复计划；	
1 2 . 9 保密措施	
1 2 . 9 公司保密制度	
1 2 . 9 项目保密制度	
1 3 技术支持与售后服务方案	
1 3 . 技术支持与售后服务体系	
1 3 . 1 技术支持与服务原则	
1 3 . 1 . I2S O 9的服务规范	
1 3 . 1 服务工作流程	
1 3 . 2 技术支持与服务体系组织保障	73
1 3 . 3 服务体系	73
1 3 . 4 技术支持与售后服务质量保障	
1 3 . 5 技术支持与售后服务内容	
1 3 . 5 售前技术服务	
1 3 . 5 售中技术服务	75
1 3 . 5 . 2 工程实施	
1 3 . 5 . 2 项目管理	
1 3 . 5 . 2 试运行阶段	
1 3 . 5 . 2 系统推广阶段	
1 3 . 5 . 2 技术文档	
1 3 . 5 . 2 技术咨询	
1 3 . 5 . 2 质保期	
1 3 . 5 售后技术服务	
1 3 . 5 . 3 技术支持热线、传真及邮件服务	
1 3 . 5 . 3 技术支持网站	

1 3 . 5 . 3 实时技术支持 .....	
1 3 . 5 . 3 对运行维护的现场技术支持和服务 .....	
1 3 . 5 . 3 故障响应及排除 .....	
1 3 . 5 . 3 例行巡检 .....	
1 3 . 5 . 3 系统更新升级 .....	
1 3 . 5 . 3 系统性能评估与优化 .....	
1 3 . 5 . 3 后期技术培训 .....	
1 3 . 5 . 3 周期性现场技术支持总结 .....	
1 3 . 5 . 3 资料定期传送 /专题讨论 .....	
1 3 . 5 . 3 系统咨询服务 .....	
1 3 . 6 技术支持与售后服务流程 .....	79
1 3 . 6 故障类 .....	79
1 3 . 6 . 1 服务流程 .....	
1 3 . 6 . 1 流程目的 .....	
1 3 . 6 . 1 流程描述 .....	
1 3 . 6 . 1 现场响应时间 .....	
1 3 . 6 技术咨询类 .....	80
1 3 . 6 . 2 服务流程 .....	
1 3 . 6 意见建议类 .....	81
1 3 . 6 . 3 服务流程 .....	
1 3 . 7 紧急情况响应服务 .....	81
1 3 . 7 紧急情况定义 .....	81
1 3 . 7 紧急情况分类 .....	81
1 3 . 7 紧急情况处理流程 .....	81

## 1 软件系统架构设计

### 1 . 概要说明

系统架构主要包括应用架构和技术架构。 系统采用基于组件的标准 S O应用架构，以及按照 S O方法构建的基于 J 2 标准的技术架构。

系统的应用架构采用了基于服务的体系架构的策略与方法， 从组件、子系统以及门户三个层次对系统进行构建， 组件组装形成子系统， 子系统集成形成门户。 门户为人员等提供一个优化的以人为中心的操作界面， 用户可以方便地对 x x的整个生命周期进行管理； 同时系统管理维护人员也可以方便地通过 p o r 对系统进行监控和管理。

系统的技术架构同样也是基于 S O方法和策略进行构建的， 它支持客户端和服务端同步和异步的两种不同的通信方式， w e层和服务层进行相对分离，支持分布式和集中式部署两种方案，并且不局限于某一种应用服务器和数据库服务器产品。

### 1 . 系统特点

#### 1 . 2 根据优化流程开发

根据流程特点进行功能设计， 采用先进的工作流引擎机制。 保证了业务功能的实现。 同时达到了灵活配置。松散耦合的目的。保证系统能够能够与原系统灵活切换。符合以 “ x x生命周期为主线 “ 的高效处理流程。使统一设计，灵活接口。

#### 1 . 2 充分利用现有资源

充分考虑现有硬件分散、系统相对独立、数据库数据分离的现状。采用分布式部署，统一数据规范、统一接口规范的设计思路，在保证系统功能灵活配置，满足业务需求的前提下，充分利用现有数据及硬件资源。

#### 1 . 2 先进的设计理念

采用国际通用的 J A语言开发，海量数据库选型、高效稳定的中间件处理。先进的 S O A架构设计，满足现有的性能需求，做到架构和系统的先进性和强大的扩展能力。 采用先进的 W e b 技术Q 做到界面简洁、易用。

#### 1 . 2 开放式的可扩展性

系统分部署式部署，子系统统一规划，即满足了分布应用的要求，又实现了统一标准。形成了统一、强大的 x x工作平台。

## 1.2 与现有系统轻松衔接

设计时充分考虑现有系统现状，开发过程和现有系统数据、应用分析同步进行，保证新系统与现有系统顺利衔接。

## 1.2 可信赖的高可靠性

考虑到实时运行，提供业务流程对可靠性的较高要求，在系统设计中充分考虑了减少和避免故障的可能和隐患，配合合理的系统部署方式和高效的维护服务，能够满足需求中对系统故障时间、修复时间和单点故障隐患的可靠性要求。

## 1. 总体体系架构

### 1.3 基于组件的 S O系统应用架构

系统的应用架构是系统进行构建的主要思路和方法，我们建议 x x系统采用基于组件的 S O的系统应用架构对系统进行构建。系统按照 S O的方法把系统从总体上划分为3个层次，分为：组件层、系统层、集成层。

- a ) 组件层 :组件层主要包括系统开发需要用到得各种组件，又可以分为横向通用组件、纵向通用组件和纵向专用组件。横向组件是大部分系统都需要用到的通用的组件，如：W e组件、日志管理、数据校验、邮件管理、打印组件、报表组件、文档管理、参数管理、单点登陆等，横向组件的作用是更好的管理和复用系统的通用组件；纵向通用组件包括在领域应用中通用的组件，如： workflow、报表工具、规则引擎、用户权限管理等领域应用中使用较为广泛；纵向专用组件是针对每一个领域专用的具有领域特色的组件，在 x x系统中纵向专用组件可以分为申请、受理、收费组件、分类组件、保密组件等等有关于 x x的组件；
- b ) 系统层：系统层包括了有组件组装得到的各个应用系统，又可以分为核心层、综合业务层和辅助管理层。核心层是整个系统的重点和难点，是整个系统最重要的组成部分，如 x x子系统是将申请人的申请进行接受和汇总子系统；
- c ) 门户平台：基于以人为本的原则，基于 p o r 技术，对系统层各个子系统进行集成。使用门户平台，用户不需要登陆每一个子系统进行相应的工作，而是在统一的门户平台进行工作。结合 workflow 技术，对于每个登陆系统的人都提供简洁统一的工作选项，对于申请人、审核人、系统管理员、维护人员、局领导等都能做到方便的操作系统，快速进行业务处理和系统管理。下图为基于 S O的 x x系统的应用架构总体设计图。

通用以上的阐述，可以看出，系统整体都是基于 SO 架构进行设计的，主要体现在如下四个方面：

- a ) 系统基于 SO 的以服务为中心的思想和方法，对 x x 系统的整体体系架构进行设计，建立了分层的松耦合、跨平台的系统架构；
- b ) 在组件层，我们采用了基于 SO 的组件模型，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。接口是采用中立的方式进行定义的，它应独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种各样的系统中的服务可以以一种统一的通用方式进行交互；
- c ) 系统采用了基于 SO 的分类集成方法对系统的业务以及服务进行分类和集成，做成统一的接口，面向业务和服务编写，以适应 SO 系统的统一交互；
- d ) 将每一种业务构成都分解成不同的组件或者子系统，将组件和子系统分开编写达到每项组件和子系统都能做到相互无关，如果一项组件和系统改变将对系统中的其余组件没有任何影响。实现组件相互之间低耦合的机制，最大程度上降低了系统的升级、业务变更对系统的影响。

同时，基于 SO 的系统应用架构具有强大的系统的扩展性：

- a ) SO 的一个中心思想就是使得企业应用摆脱面向技术的解决方案的束缚，轻松应对企业商业服务变化、发展的需要，本方案很好地体现了 SO 的这一中心思想；
- b ) 工作流和业务规则引擎的采用极大地提高了系统对于业务流程和规则变化的适应性。工作流引擎可以使得在业务流程发生变化时使得系统调整最小，而不需要向传统的需要完全重新开发；业务规则引擎的采用使得业务规则发生变化时只需对业务规则进行重新描述即可完成系统的转换。
- c ) 组件模型、组件集成技术的采用使得系统在进行业务功能的调整时，可以把变化局限于某一个范围之内，在需要时还能进行灵活的替换。由于系统应用架构是根据每一项业务或者流程编写所以对于系统的扩展非常方便，只要对新加入的业务对应加入新的组件就可以实现对 SO 系统的扩展；

总之，本节提出的基于组件的 x x 系统完全体现了 SO 的核心思想，通过分层组件规划、集成、工作流引擎、业务规则引擎等方法和技术充分体现 SO 的策略与方法，并且很好地实现系统的可扩展性、可移植性等等。

### 1.3 系统技术框架

x x 系统基于 J2 规范实现，整个架构建立在 S t r 框架、S p r 框架和 D A 模式基础之上，并提供了对于 E J B W e S e r v i c e s 组件技术的集成机制。技术框架逻辑上可



分为：客户层、W E层、业务层、持久层、资源层、核心层。如下图所示为系统的技术框架。

客户层：客户端计算机的浏览器，用于展现页面。

W E层：W E层基于 S t r u t s, 完成转发请求、H t 请求合法性校验、H t 请求参数与数据传输对象 D T 间的绑定、H t 请求参数有效性校验、用户操作权限检查、记录用户访问日志、显示系统运行异常等任务。

业务层：业务层基于 S p r 框架, 完成业务数据校验、业务逻辑处理、事务管理、记录业务处理日志、抛出业务处理异常等任务，同时它也支持 W e S e r v i s e J 等组件服务模型。

持久层：持久层基于 D A 进行构建，完成数据读取、数据存储、封装 S Q 异常、抛出 S Q 异常、记录数据读写日志等任务。

资源层：资源层包括数据库服务器、X M 存储文件等，是数据永久存储的介质。

核心层：核心层表现为系统提供的基础类库，为 W E层、业务层和持久层提供支持。包括日志记录组件、异常处理组件、事务处理组件、I o 容器封装组件、W E层数据绑定组件、W E层数据校验组件、权限检查组件、持久层辅助组件、其他开源项目类库组件等。

本技术框架的特色或优势主要体现在如下几个方面：

- ( 1) 系统技术框架提供了对 S O 的完整支持；
- ( 2) 对于同一个应用系统，系统同时支持集中式和分布式两种部署方案，系统采用分离 U 层和 B 层的方式来实现分布式的实现；