

【软考达人】

软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



微信扫一扫，立马获取



6W+ 免费题库



免费备考资料

PC版题库: ruankaodaren.com

摘要：

2012年1月，我作为项目经理，主持XX保险公司全国再保险大集中管理系统的建设项目，该项目为期2年半，总投资为1800万人民币，通过该项目，实现XX保险公司整体信息化转型升级的战略中再保险板块的落地，完成全国海量再保险业务数据的集中部署运行，迁移整合历史数据，全面替代上一代系统。该项目时间紧任务重、涉及人员组织多，直接相关XX保险公司内部60个部门400余人，外部配合协作30多个厂商团队300余人。该项目2014年5月完成系统上线，2014年6月通过最终验收，得到了用户的一致肯定，顺利达成了项目既定目标。本文作者结合实际经验，以该项目为例，讨论一下项目建设的【软件分析、软件设计、】这几个过程来进行论述。

正文：

2012年1月，我作为项目经理，主持XX保险公司全国再保险大集中管理系统的建设项目，该平台为期2年半，总投资为1800万人民币。该项目时间紧任务重，具有相当的挑战性，一是业务模式升级，需按照最新的再保险业务流程，完成系统功能的分析开发，进而具体落地公司再保险业务流程的再造；二是技术要求高，要实现全国海量再保险业务数据的集中部署运行，每日处理数据量达到3000万笔以上，同时要满足性能要求。三是数据整合难，需要将上一代系统的中历时十年的数据，按其有效性进行分类、转化、整合，实现历史存续业务数据在新系统环境下，按照新新模式正常运行。四是涉及人员组织多，直接研发团队36人，XX保险公司总部再保险部、财务部、风险部、八大业务部、40个省公司等400余人，同时涉及外部配合协作承保系统、核保系统、理赔系统、收付费系统、财务系统等30多个厂商团队300余人。我担任项目第一负责人，负责**项目整体技术方案**评估、立项论证以及项目管理工作。在项目启动前，负责分析项目的预期经济效益、可选技术方案，分析关联项目影响，并向公司提交立项报告。项目启动后，作为主要负责人，牵头与公司内部技术专家、外部架构师一同建立项目技术架构组，设计项目整体技术架构，同时挑选项目内部成员，建立需求分析组、系统开发组、系统测试组、运维支持组，开展业务需求分析、系统设计、数据迁移方案、上线切换方案工作。一方面，我个人接受各组工作汇报，指导团队研发工作，监控整体工作进度。同时，我及时向公司领导、项目客户方、相关项目团队汇报沟通工作进展、阐明关键技术要点。

该项目 2014 年 5 月完成系统上线，2014 年 6 月通过最终验收，得到了用户的一致肯定，顺利达成了项目既定目标。**尤其在系统整体技术架构方面，效果尤为突出，一是保持了较好的灵活性，提供了较多的基础组件，使得应用功能便于横向扩充，满足了项目业务灵活性；二是系统性能整达到各个层次的整合设计，数据库层（分片、读写分离）、应用开发层（缓存技术、查询条件的严格限制），服务部署层（采用集群部署、F5 分发），使得业务日处理能力 5500 万单以上，预留了 80%性能，满足了公司未来 3-5 年的业务增量。**

以该项目为例，就**系统架构设计的 XXX 方面**进行讨论，具体从规划 XXX、管理 XX 和控制 XX 及 XXXX 管理这几个过程来进行论述。

论文主干，出哪个方面就写哪个方面（每一列为一个方面），里面每个点至少写 1-2 段，凑够 1500-1800+字，重点突出“我”作为项目架构师的思考和实践。

性能设计	软件架构 (质量) 评估	软件设计模式	数据库访问层设计	微服务架构
1 性能设计是一个整体，必须有整体的一致性，避免“大河接了小水管”的片面设计。 如何做到整体： 1.1 充分理解需求 1.2 梳理运行环境 1.3 全流程分析 1.4 分层模型 2、性能设计要抓住关键点—风险点 2.1 哪些地方	1、质量的要点 1.1 性能 1.2 可靠性：出错情况下的容错能力 1.3 可用性：正常/异常时间比 两次异常间隔时间 1.4 安全性 1.5 可修改性 1.6 功能性 1.7 可变性 1.8 互操作性 2 采用的评估方法+实施过程+实际效果。	常见设计模式以及作用 1 创建型：对对象实例化过程进行抽象。 1.1 抽象工厂 Abstract factory 1.2 建造者 builder 1.3 工厂 Factory method 1.4 原型 prototype 1.5 singleton 2、结构性模式 组合类和对象，获得更大的结构。一般使用继承、封装、代理等。 2.1 适配器 adaptor 2.2 bridge 桥接 2.3 组合 composite 2.4 装饰 decorator 2.5 外观 façade 2.6 享元 flyweight	一、五种访问模式 1.在线访问。业务之间访问数据，无数据库访问层 2. dao。将底层数据库与高层业务逻辑分开，访问特定数据源 3 dto：一组对象组成的数据容器。跨进程或者跨网络传输数据 4 离线数据模型：从数据源获得数据后，按照预定结构存放在系统中的数据。往往与 XML 集成使用 5 对象/关系映射	一、微服务的好处 1 实现组件化，单个服务实现简单，能够聚焦一个指定的业务功能或业务需求。 2.功能明确，易于理解。小团队能够更关注自己的工作成果。 3. 围绕业务功能构建开发团队。更符合企业的分工与组织结构 4、支持多种语言和平台（采用 HTTP 等通用、轻量协议交互）进行互操作。支持不同平台。 5、离散化数据管理。

<p>是风险点。</p> <p>2.2 哪些地方是敏感点，</p> <p>2.3 哪些地方是权衡点</p> <p>3、要敢于突破</p> <p>3.1 特定环境要敢于突破。比如必须用接口？比如不准用存储过程。</p> <p>4、要有效地验证</p> <p>4.1 评估推算</p> <p>4.2 模拟验证</p>	<p>3 选 1 来说。</p> <p>2.1 SAAM</p> <p>2.2 ATAM 体系结构平衡法</p>	<p>2.7proxy 代理</p> <p>3 行为型</p> <p>用于对象之间的职责以及提供服务的分配。描述对象之间如何相互协作。</p> <p>3.1 职责链:chain of responsibility</p> <p>3.2 command 命令</p> <p>3.3 解释器 interpreter</p> <p>3.4 iterator</p> <p>3.5 mediator</p> <p>3.6 memento</p> <p>3.7observer</p> <p>3.8state</p> <p>3.9strategy</p> <p>3.10Template method</p> <p>3.11 visitor</p> <p>4\ 实际使用的场景以及取得的效果。</p> <p>效果有：</p> <ol style="list-style-type: none"> 1. 简化并加快设计 2. 方便开发人员之间的通信：-- 前端逻辑和后端业务处理之间等 3. 降低风险 4. 有助于转到面向对象 	<p>ORM。</p> <p>ORM 可以是一种工具或者平台，将程序中的数据映射成关系数据库中的记录，或者反之。使得开发面向对象，简化数据库开发相关工作。</p> <p>二、实际使用效果。</p> <p>1、哪些场景实际用了什么设计方法，为什么，如何设计的，取得的效果如何。</p> <p>在业务逻辑部分，采用 orm 模式。</p> <p>在特定的高性能数据提取要求情况下，采取了之间读取数据库（及个边）</p> <p>在系统数据内部自运算环境，采用了个别存储过程。以保持效率。</p>	<p>无法创建或维护统一的数据模型，需要进行数据模型的离散化管理？</p> <p>6 基础设施自动化。通过持续集成工具实现基础设施自动化。</p> <p>二、实际使用情况。</p> <p>1、使用已经有的微服务基础设施：</p> <p>用户登录认证服务</p> <p>用户权限控制服务</p> <p>主数据集合服务</p> <p>接口发现服务。</p> <p>构建再保险微服务：</p> <p>一是先分保后出单模块</p> <p>二是自动分保模块</p> <p>三是账务处理模块</p> <p>四是结算管理</p> <p>五是资信管理</p> <p>主要遇到的问题：</p> <p>1、微服务不是越多越好，需要充分理解用户需求、实际业务场景的前提下，做出合理划分归集</p> <p>2、基础设施的要求更高</p>
--	--	--	--	--

				3、信息安全挑战加大。

经过我和团队的不懈努力，历时一年，项目终于于 2014 年 6 月通过顺利通过了验收，并得到了一致好评，运行至今，用户反馈良好，XX 保险公司的再保险业务运营水平得以提升。但是，在实施过程中，也暴露了一些具体问题，例如跨系统之间接口交互时，由于业务复杂，简单的队列机制无法满足繁忙场景，需要建立具有动态优先级调整机制的处理队列等等，这些问题通过应急处理和协调，以及高层领导的推动，都得到了妥善解决，没有影响到项目的总体进度。我们已经把这些经验和教训，总结到了工作总结里面，向其他技术人员分享，为今后系统架构设计提供帮助。相信通过不断持续学习改进，加强自己的系统分析与设计能力，努力工作提升工作水平，为社会和公司多贡献一点自己的价值。