全国计算机技术与软件专业技术资格（水平）考试

系统架构设计师《论文写作一本通》

软考诸葛老师

本资料使用说明：

本资料为诸葛老师系统架构设计师论文写作一本通，包含复习说明、万能模板、论文素材、论文范文、历年真题等全部内容，请同学们务必结合论文专题课程认真学习。

第一章是论文整体分析，大家认真浏览一遍，心中有数即可。

第二章是搭建自己的万能模板，这里会将 2500 字的论文拆解开来，其中会有 1300 字是可以提前准备好的，包括摘要+项目背景+项目结尾，同时提供一个万能模板，大家仿照里面说明选择好自己的项目，并写好自己的万能模板。

第三章是正文写作部分，除万能模板外，还需要再写 1200 字左右，这部分是论文核心，需要依据具体论文题目来写，我们列举出了历年常考的，出题概率大的一些论文题目，并给出分析步骤和范文供大家参考练习。

第四章是历年论文真题，大家浏览一遍历年真题和写作要点，尤其是子题目的问法，看哪里还不会。

2023 年 11 月机试改革之后，架构师论文仍然是需要写摘要+正文的，分别有不同的输入框，直接打字即可，机试更加利好论文，可以随意修改重构，时间也足够。

淘宝 APP 扫码购课	微信扫码刷题	微信公众号	诸葛老师抖音

第 1 章 论文整体分析

1.1 论文复习说明

论文写作介绍：

考试时间：下午 14：30-16：30；考试时长：120 分钟。

考试题目：四选一，必有 1 题考查软件架构，每道题会有相关概述和子题目，要仔细审题。

考试形式：机试，约 2500 字。

论文题目类型：

1、软件架构，自从 2020 年以来，更偏向于考查具体的架构，而不是之前的宏观上的架构风格、架构评估等内容。尤其喜欢考查改版后的八大架构，如云原生、微服务、安全架构等。

2、系统开发，这块也是每年必考，软件工程全生命周期都有可能考查，而且也是更具体，比如具体的开发方法、开发模型，以及需求分析、设计、测试、运维等全过程。

3、系统可靠性、安全性、容错技术等。

4、企业应用集成、企业集成平台等。

5、其他：项目管理、数据库等。

论文复习方法：

1、零项目经验的学员不要担心，听诸葛老师论文写作专题课程，掌握架构师论文写作技巧，参考优秀范文改写成自己的，完全可以通过考试的。

2、写论文的前提是已经完成所有基础知识课程的学习，建议在考前一个半月开始准备论文，注意以下事项：①论文写作专题课程的学习；②确定自己要写的项目，并确定论文模板；③根据确定的项目和模板，按照老师给定的写作顺序准备论文，建议每周写一篇论文，至少完成 5 篇论文（在电脑上撰写）。

教材第 20 章系统架构设计师论文写作要点

此章节中，所述关于论文写作的准备工作、论文格式、解题步骤、注意事项、分段写法、摘要模板等，均和一本通的内容大同小异，同学们按照一本通和论文专题课程复习即可。

1.2 历年真题汇总

此表仅包含论文大题目，以供整体分析。每年真题的子题目，也有必要了解，见第四章。

年份	试题一	试题二	试题三	试题四
202411	论面向服务的架构设计	论软件维护及其应用	论多源异构数据集成方法	论分布式事务及其解决方案
202405	论大数据 lambda 架构	论模型驱动架构设计方法及其应用	单元测试	论云上自动化运维及其应用
2023	可靠性分析与评价方法	面向对象分析	多数据源集成	边云协同
2022	论基于构件的软件开发方法及其应用	论软件维护方法及其应用	论区块链技术及应用	论湖仓一体架构及其应用
2021	论面向方面的编程技术及其应用	论系统安全架构设计及其应用	论企业集成平台的理解与应用	论微服务架构及其应用
2020	论企业集成架构设计及应用	论软件测试中缺陷管理及其应用	论云原生架构及其应用	论数据分片技术及其应用
2019	论软件设计方法及其应用	论软件系统架构评估及其应用	论数据湖技术及其应用	论负载均衡技术在 web 系统中的应用
2018	论软件开发过程 RUP 及其应用	论软件体系结构的演化	论面向服务架构设计及其应用	论 NoSQL 数据库技术及其应用
2017	论软件系统建模方法及其应用	论软件架构风格	论无服务器架构及其应用	论软件质量保证及其应用
2016	论软件系统架构评估	论软件设计模式及其应用	论数据访问层设计技术及其应用	论微服务架构及其应用
2015	论应用服务器基础软件	论软件系统架构风格	论面向服务的架构及其应用	论企业集成平台的技术与应用
2014	论软件需求管理	论非功能性需求对企业应用架构设计的影响	论软件的可靠性设计	论网络安全体系设计
2013	论软件架构建模技术与应用	论企业应用系统的分层架构风格	论软件可靠性设计技术的应用	论分布式存储系统架构设计
2012	论企业信息化规划	论决策支持系统的	论企业应用系统的	论基于架构的软件

	的实施与应用	开发与应用	数据持久层架构设计	设计方法及应用
2011	论模型驱动架构在系统开发中的应用	论企业集成平台的架构设计	论企业架构管理与应用	论软件需求获取技术及应用
2010	论软件的静态演化和动态演化及其应用	论数据挖掘技术的应用	论大规模分布式系统缓存设计策略	论软件可靠性评价

1.3 论文写作原则

基本原则：不能口语化、每段不宜太长；字数一般在 300+2200 左右，机试时有实时字数统计，多注意把握；写正文时，不要生硬的回答问题，要根据问题要点组合成一篇通顺的文章；论点分开论述，没必要全写编号。

其他原则：提前准备好自己要写的项目，必须是近三年的中大型商业项目；不要猜题，要复用构件(摘要+项目背景+结尾)，不要整篇复用；不要直接抄范文，要选择适合自己的范文进行范文修改，改写成属于自己的论文；练习论文主题时发现有不会的知识点，一定要全部背会；站在系统架构设计师的角度看待项目，技术细节适中，理论联系实际；不同角度看论文，以论文写作技巧课程为依据，对自己的论文进行自评；最后，要勇于迈出第一步，万事开头难，坚持写完第一篇，之后越写越容易。

1.4 论文常见问题

问题 1:论文字数范围，各部分字数如何分配?

答：建议摘要 300 字左右，正文 2000-2700 字之间，建议 2200 字；具体建议：摘要 300 字+项目背景 500 字+正文主体 1200 字+结尾 500 字。

问题 2:是否需要写论文题目?

答：机试改革后，直接在考试软件里选择题目即可，不需要再写论文题目。

问题 3:我没有当过公司的系统架构设计师怎么办?

答：如果你是专业技术人员，即使不是系统架构设计师，但也做过项目，那就站在系统架构设计师的角度去写熟悉的项目；如果你是完全零基础，那就参考我们的优秀范文，选择一个适合自己的项目进行改写。

问题 4:我做的是公司内部的项目，没有客户怎么办?

答：当做正常的商业项目撰写即可。如某年某月，公司要做什么项目，任命我为系统架构设计师，最后收尾仍然要写客户一致满意，切记。

问题 5:项目背景如何构造才比较合理?

答：关于项目时间：建议使用离考试三年内，持续时间 6 个月以上，并且已完成的项目；如果项目真的比较久远，或者持续时间较短，但是规模却是足够的，那么可以根据实际情况灵活修改时间。

关于项目名称：真实项目名称不需要写，也不能写，只能写某地级市、某医院、某公司等。

关于项目级别：建议是省市级、集团级，不要是县级、社区级等。如果你做的项目规模是符合要求的，真的是县级的真实项目，那也没有问题。

关于项目团队成员：可省略。为了保证真实性，也可以简单带过，例如有需求分析人员 3 人，开发人员 10 人，测试人员 6 人等。

关于项目金额：要和项目时间还有项目团队成员相对应，如果项目组一共有 30 人，按每人每月 2 万算，一个月需要 60 万，再乘以持续时间（假设为 15 个月），那么项目金额在 900 万左右是比较合理的，但不要写整数，可以写 869 万、915 万等字眼。

特别注意：如果是自己改编的项目，请不要构造太大的项目，项目金额最好在 500 万左右。

问题 6:如何记住提前准备的论文?

答：一定要自己撰写，自己写过，才能更好地记住。建议文章采用统一的模板和结构，搭建好自己的模板。正文部分按论文题目要求去写，一定要回应子题目。除此之外，一定要自己动手撰写练习，在要求的两个小时内撰写完毕，直接在电脑打开一个 word 文档撰写即可。

1.5 论文准备顺序

最终论文押题要到考前一周左右才能出来，这时才准备论文肯定来之不及，因此提前准备是完全有必要的，准备顺序如下：

1、请同学们直接按照第 3 章的标题顺序来依次准备论文。本标题顺序包括了历年常考的论文题目，并按重要程度和频率排序。

2、建议至少准备到第 7 篇，后面的论文，也建议浏览下范文和相关知识点，如果考到不至于措手不及。

3、7 篇论文不多哈，当你有了自己的模板，写了三篇论文之后，后面再撰写起来就很顺手了。

注意：考前一周左右我们会发布论文押题题目，请耐心等待，不要急着催。押题题目一般就在第 3 章的论文题目里，所以大家一定要提前做好论文准备。

第 2 章 搭建自己的万能模板

请大家在学完本章节后，按步骤一步步地搭建好自己的模板，后面也有万能模板供模仿。如果是零基础学员，不知道怎么选择项目，可以提前看第 3 章中提供的范文，从中选一个修改成自己的项目。有自己真实项目经历的同学，尽量用自己的真实项目，如果自己项目规模不够，可以在真实项目基础上去扩展，因为范文的项目可能会被很多人采用，容易导致重复。

2.1 选择合适的项目

选择中大型商业项目，一般金额在 200 万以上，研发周期在 6 个月以上的项目。

（1）推荐项目

政府或大型信息系统项目：各国企、事业单位、军方、医院、银行、股份公司大企业的 OA、ERP；大数据、云计算等各种软件及信息系统。

（2）不能选项目

小型企业项目：如进销存系统、图书管理系统、单机版系统。

尚未完成的项目：一定要已经完成并上线运行，并最好是三年内的。

纯建网站项目：如建设企业或政府门户网站，只有静态网页链接介绍，没有后台大型应用的。

硬件项目：如综合布线、安防、视频会议等。

纯技术项目：如数据升级迁移、内部技术研究等。

（3）建议

在推荐的范围内有限选择自己做的或熟悉的中大型项目，如果是零基础没有做过项目的同学，就参考后面的范文，用别人的项目来模仿。

2.2 论文写作要点

1.论文摘要

摘要是对论文全文内容的浓缩和精华，通过摘要，可以让读者迅速总揽全文内容。一般的，摘要至少包含三部分内容。一是项目背景简介，如时间、项目名称、项目主要内容；二是作者角色及工作内容；三是项目技术简介，如采用的主要方法、结论和成果。摘要字数为 300 左右为宜。

2.项目背景

项目背景，500-600 字。

①项目由来/缘起/定位/目标，主要介绍项目的前提和诞生的背景等；

②项目主要内容：简要介绍，不是摘要中简单的重复，应比摘要稍详细；具体可以参考下述 5w2h 框架：

●when：何时，近三年项目（体现技术先进性）；

●where：何地，某省/某市，注意不能实名；

●who：甲乙方、作者，甲方名称不能实名，我方称“我司/我单位/我厂/我公司”；

●why：为何立项，项目建设目的；

●what：项目名、项目内容、作者的工作内容；

●how much：项目预算，不宜太小，不能太大；

●how：作者采用的技术/方法；

注意 5W2h 内容及先后次序可根据具体情况而定，没有的内容可以不要写。

3.论文正文

1）理论部分，400~500 字

①紧扣题干要求：一般是对题干的应答，注意要写成一段；

②逐点回答：分论点的基本概念、基本原理、应用场景、简单举例即可；

③惜字如金：注意控制字数，不要挤占实践部分。

2）实践部分，900~1100 字

①结构上：与理论部分相呼应对应，最好保持一致

②分论点标题：最好拟一个小标题，注意小标题的行文格式

③分论点内容：先识别问题，阐述 why；然后分析/设计/解决问题，阐述 how；最后阐述效果。

4.论文结尾

结尾，400-600 字。

①项目效果：呼应论点、上线、稳定运行、获得好评、下一步计划等；

②存在的问题：阐述小问题，且已解决的；

注意各部分之间要有过渡，不能太生硬。

2.3 提前准备论文摘要

摘要可以参考范文或者下面的格式，根据自己选择的项目准备一个就可以了。建议逻辑上分两层意思，第一层是通用的介绍项目背景；第二层是根据不同的论文题目进行发挥，简单回应子题目并介绍论文结构。

包含内容：项目名称、项目金额、项目历时、项目简介、我的责任；本文主题概括(具体可以参考后面的万能模板)。

论文摘要格式参考：

(1)本文讨论……系统项目的……(指的是项目主题，例如进度管理等)，该系统是由某单位建设的，投资多少万，系统是用来做什么的(项目背景，简单功能等)。在本文中，首先讨论了……(过程、方法、措施)，最后……(主要是不足之处/如何改进、特色之处、发展趋势等)。在本项目的开发过程中，我主要担任了……(在本项目中的角色)。

(2)根据……需求(项目背景)，我所在的……组织了……项目的开发。该项目……(项目背景、简单功能介绍)。在该项目中，我担任了……(角色)。我通过采取……(过程、方法、措施等)，使项目圆满成功，得到了用户的一致好评。但通过项目的建设，我发现……(主要是不足之处/如何改进、特色之处、发展趋势等)。

(3)...年...月，我参加了....项目的开发，担任……(角色)。该项目投资多少，建设工期是多少，该项目是为了(项目背景、功能介绍)。本文结合作者的实践，以……项目为例，讨论……(论文主题)，包括……(过程、方法、措施)。

(4)……是……(“戴帽子”，讲述论文主题的重要性，比如进度的重要性)。本文结合作者的实践，以……项目为例，讨论……(论文主题)，包括……(过程、方法、措施)。在本项目的开发过程中，我担任了……(角色)。

摘要应该概括地反映正文的全貌，要引人入胜，要给人一个好的初步印象。一般来说，不要在

摘要中“戴帽子”如果觉得字数可能不够，例如少于 300 字，则可适当加 50 字左右的帽子。

上述的“技术、方法、工具、措施、手段”就是指论文正文中论述的技术、方法、工具、措施、手段，可把每个方法（技术、工具、措施、手段）的要点用一两句话进行概括，写在摘要中。

在写摘要时，千万不要只谈大道理，而不牵涉到具体内容。否则，就变成了“摘要中没有实质性内容”。

2.4 提前准备项目背景

项目背景及过渡部分建议 500-600 字左右，不能超过太多，需要提前准备好，尽量通用化，不要和论文主题相关，这样考试时候无论什么论文主题都可以直接默写，只需要在最后写一段过渡语句，过渡到下一个论点。

包括内容：项目开发的原因、你的岗位职责、项目开发周期及规模、项目功能组成介绍、项目技术(可省略)。

具体可以参考后面的万能模板来写。

2.5 论文正文写作

正文应该按照论文题目和子题目的要求来写作，并且一定要回应论文子题目。

正文需要写 1200 字左右。正文部分不在万能模板里，需要根据不同的题目来准备，详见第 3 章内容，本章可以先略过，先准备好自己的万能模板。

2.6 提前准备结尾

结尾也很重要，从本质上说，是让我们总结项目收获与不足的，另一方面，也是我们整体补救论文的最后一步。如果最后发现字数还不够，结尾就要多写一些，如果字数多了，结尾就要少写一点，如果文章前面写的很差，结尾就一定要多重视，多揣摩揣摩。

结尾可以写 400-600 字左右，是对整体论文的概括。包括：项目上线及运行效果、客户评价、项目收获、项目不足和解决思路。

具体可以参考后面的万能模板来写。

2.7 论文万能模板

按照上面提到的步骤，我们可以总结出如下的套路模板，大家可以参考，一定要使用自己的项目来模仿。其中标红部分是完全复用的，任何题目都不用修改。

摘要模板(时间+项目+项目简介+投入+历时+成功交付客户好评+结合具体题目说明本文结构):

2022 年 5 月，我参与了某邮电设计院的无线网络仿真系统开发项目的建设，并担任系统架构设计师，负责系统架构设计工作。该系统包括无限信号仿真、网络流量模拟以及动态环境建模三大功能模块，能够重现复杂的真实世界场景，包括密集城区、开阔地带和障碍物影响等，从而在虚拟环境下对无线网络进行全面测试和优化。该项目总投入 326 万元人民币，历时 15 个月，于 2023 年 8 月正式交付运行至今，受到了客户的一致好评。本文结合笔者的实际工作经验就该项目的……(根据不同论文题目去简要概括本文内容)。

项目背景模板(为什么要做这个项目+项目功能和技术介绍+回应子题目并过渡到主体)：

随着 5G 及未来 6G 网络技术的迅猛发展，无线网络的复杂性和多样性对传统测试手段提出了前所未有的挑战。传统实验室测试往往受限于物理设备的数量和成本，难以覆盖所有可能的网络场景和极端条件，这可能导致网络设备在实际部署中遭遇未预见的问题。鉴于此，该邮电设计院决定使用技改经费投资建设一项无线网络仿真系统开发项目，以纯软件的方式模拟无线网络设备，在虚拟平台中对无线网络设备进行全面测试与验证。

我所在的企业成功中标该项目，并于 2022 年 5 月正式启动该项目的建设工作，我被任命为该项目的系统架构设计师，负责系统架构设计工作。该项目总投入 326 万元人民币，建设周期从 2022 年 5 月 15 日至 2023 年 8 月 31 日止，历时 15 个月。为了实现高性能和高效率的仿真，我们选择了两台联想 ThinkStationP510 工作站作为软件运行平台，使用 C 语言模拟 CPU 指令集及外部设备驱动，使用 LabVIEW 图形化语言搭建网络设备配置终端界面，打造兼具功能性和易用性的仿真系统。项目的核心建设内容围绕三大模块展开：一、无线信号仿真：通过软件模拟无线信道特性，包括多径传播、干扰、衰落等，以验证无线设备在复杂电磁环境中的表现。二、网络流量模拟：创建各种网络流量模式，包括数据包的生成、传输和接收，用于测试网络容量、延迟和丢包率等关键性能指标。三、动态环境建模：构建动态变化的网络拓扑和移动场景，如车辆网络、无人机群组等，以评估网络在不断变化条件下的鲁棒性和适应性。

笔者所在的企业尽管在过往的无线网络仿真项目中积累了丰富的经验，但每个网络标准和技术的更新迭代都带来了新的挑战。本项目需针对特定的 5G 商用场景进行定制化开发，同时严格遵守行业标准和安全规范，加之对敏感信息的保密要求，无疑增加了项目实施的复杂性和难度。于是笔

者决定在……(过渡段，可以自行参考论文来写，需要回应子题目并引出正文)。

项目总结模板(强调项目顺利交付、运行反馈好+自己的收获或不足之处):

经过近 15 个月的项目研发，该无线网络仿真系统顺利投入使用，协助客户实现对无线网络设备进行全面功能和性能上的测试，运行至今客户反馈良好。该系统由于性能要求高，技术实现难度高，项目建设周期长等原因，建设过程困难重重。但由于笔者及项目团队成员十分重视项目的……(回应具体论文题目)，最终保证了该项目按质按量顺利交付。

当然，在本项目中，还有一些不足之处，比如：……(自己去想一些小问题，切忌，别出现什么大问题)，不过，经过我后期的纠偏，并没有对项目产生什么影响。在后续的学习和工作中，我将不断的充电学习，和同行进行交流，提升自己的专业技术水平，更好的胜任系统架构设计的工作。

第 3 章　正文素材及范文

根据第 2 章，我们已经准备好了自己的论文模板，明确了自己的项目选题、项目摘要、项目背景以及结尾。那么，本章节我们来看如何写正文部分。正文部分需要写 1200 字左右，包括两块：

①根据论文题目写相关内容。这一块是我们提前准备的重点，不能只列举理论，一定要根据实际项目举几个真实的例子。

②回应子题目。一般可以直接按子题目顺序写论文正文，重点突出过程，可以参考历年真题的子题目是怎么问的。

本章将针对常考的论文题目，给出写作思路和范文参考，在学习本章内容时，一定要重点关注正文部分的写作，其他项目背景等模板部分使用自己准备好的万能模板即可。

特别注意：

①范文仅供参考，正式考试时请勿照抄，一定要自己改写。

②范文可能会出现字数超标、与课程讲的不太一样等，这都是正常的，主要是使同学们掌握写作的思路。

3.1 第二版下篇具体应用架构

纵观历年论文真题，我们可以发现，近几年关于纯理论的题目的考查是越来越少了，如架构风格、架构评估这些是 19 年及之前常考的，而现在常考的是具体的应用架构，如云原生架构、微服务架构、企业集成架构、系统安全架构等。尤其是在第二版教材更新后，教材下篇的章节目录，分别是：信息系统架构、层次式架构、云原生架构、面向服务架构、嵌入式系统架构、通信系统架构、安全架构、大数据架构。

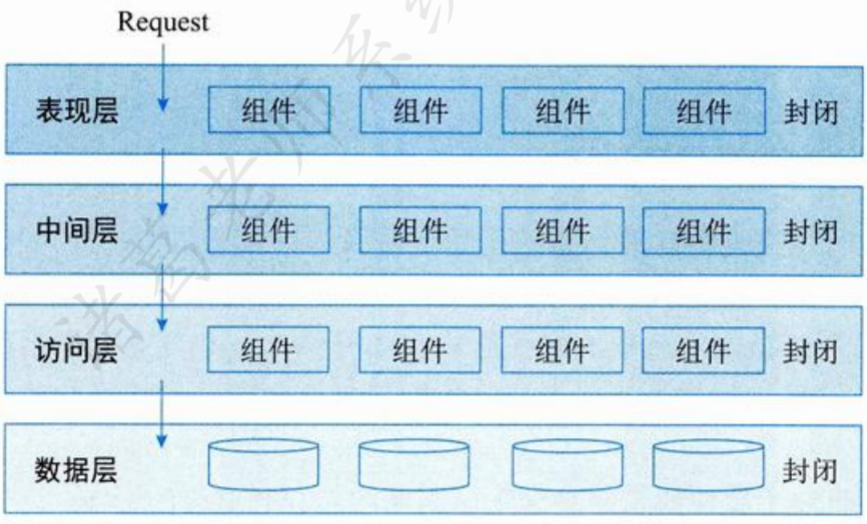
根据以上分析，架构论文的侧重点应该是第二版教材下篇的八大架构，而且 2024 上半年就考查了大数据架构。因此，诸葛老师预测，2024 下半年论文可能考查的具体架构是：层次式架构、云原生架构。

以下提供一些具体架构范文，仅供参考。

层次式架构

层次式体系结构设计是将系统组成一个层次结构，每一层为上层服务，并作为下层客户。

软件层次式体系结构是最通用的架构，也被叫作 N 层架构模式。大部分的应用会分成表现层（或称为展示层）、中间层（或称为业务层）、数据访问层（或称为持久层）和数据层。



常用的层次式架构

层次式架构范文

摘要：我所在的单位是国内主要的商业银行之一，作为单位的主要技术骨干，2022 年 2 月，我主持了远期结售汇系统的开发，该系统是我行综合业务系统的一个子系统，由于银行系统对安全性，可靠性，可用性和响应速度要求很高，我选择了三层 C/S 结构作为该系统的软件体系结构，在详细

的设计三层结构的过程中，我采用了字符终端为表示层，CICS TRANSATION SERVER 为中间层，DB2 UDB7.1 为数据库层，并采用了 CICS SWITCH 组，并行批量的办法来解决设计中遇到的问题，保证了远期结售汇系统按计划完成并顺利投产，我设计的软件三层结构得到了同事和领导的一致认同和称赞。但是，我也看到在三层结构设计中存在一些不足之处：比如中间层的负载均衡算法过于简单，容易造成系统负荷不均衡，并行批量设计不够严谨，容易造成资源冲突等。

我所在的单位是国内主要的商业银行之一。众所周知，银行的业务存在一个“二八定理”：即银行的百分之八十的利润是由百分之二十的客户所创造。为了更好地服务大客户，适应我国对外贸易的蓬勃发展态势，促进我国对外贸易的发展，2022 年 2 月，我行开展了远期结售汇业务。所谓的远期结售汇就是企业在取得中国外汇管理局的批准后，根据对外贸易的合同等凭证与银行制定合约，银行根据制定合约当天的外汇汇率，通过远期汇率公式，计算出交割当天的外汇汇率，并在那天以该汇率进行成交的外汇买卖业务。远期结售汇系统是我行综合业务系统的一个子系统，它主要包括了联机部分、批量部分、清算部分和通兑部分，具有协议管理、合约管理、报价管理、外汇敞口管理、帐务管理、数据拆分管理、报表管理、业务缩微和事后监督等功能。我作为单位的主要技术骨干之一，主持并参与了远期结售汇系统的项目计划、需求分析、设计、编码和测试阶段的工作。

由于银行系统对安全性，可靠性，可用性和响应速度要求很高，我选择了三层 C/S 结构作为该系统的软件体系结构，下面，我将分层次详细介绍三层 C/S 软件体系结构的设计过程。

1、表示层为字符终端。我行以前一直使用 IBM 的 VISUAL GEN2.0 附带的图形用户终端来开发终端程序，但在使用的过程中，分行的业务人员反映响应速度比较慢，特别是业务量比较大的时候，速度更是难以忍受。为此，我行最近自行开发了一套字符终端 CITE，它采用 VISUAL BASIC 作为开发语言，具有响应速度快，交互能力强，易学，编码快和功能强大的特点，在权衡了两者的优点和缺点之后，我决定选择字符终端 CITE 作为表示层。

2、中间层为 CICS TRANSATION SERVER(CTS)。首先，我行与 IBM 公司一直保持着良好的合作关系，而我行的大部分技术和设备都采用了 IBM 公司的产品，其中包括了大型机，由于 CICS 在 IBM 的大型机上得到了广泛的应用，并在我行取得了很大的成功，为了保证与原来系统的兼容和互用性，我采用了 IBM 的 CTS 作为中间层，连接表示层和数据库层，简化系统的设计，使开发人员可以专注于表示逻辑和业务逻辑的开发工作，缩短了开发周期，减少开发费用和维护费用，提高了开发的成功率；其次，对于中间层的业务逻辑，我采用了我行一直使用的 VISUAL AGEFOR JAVA 作为开发平台，它具有简单易用的特点，特别适合开发业务逻辑，可以使开发人员快速而准确地开发出业务逻辑，确保了远期结售汇系统的顺利完成；最后，由于采用了 CTS，确保了系统的开放性和互操作性，保证了与我行原来的联机系统和其他系统的兼容，保护了我行的原有投资。

3、数据层为 DB2 UDB7.1。由于 DB2 在大型事务处理系统中表现出色，我行一直使用 DB2 作为事务处理的数据库，并取得了很大的成功，在 DB2 数据库的使用方面积累了自己独到的经验和大量的人才，为了延续技术的连续性和保护原有投资，我选择了 DB2 UDB7.1 作为数据层。

但是，在设计的过程中我也遇到了一些困难，我主要采取了以下的办法来解决：

1、CICS SWITCH 组。众所周知，银行系统对于安全性，可靠性，可用性和响应速度要求很高，特别是我行最近进行了数据集中，全国只设两个数据中心，分别在 XX 和 YY 两个地方，这样对以上的要求就更高了，为了保障我行的安全生产，我采用了 CTS SWITCH 组技术，所谓的 CICS SWITCH 组，就是一组相同的 CTS，每个 CTS 上都有相同的业务逻辑，共同作为中间层，消除了单点故障，确保了系统的高度可用性。为了简化系统的设计和缩短通讯时间，我采用了简单的负载均衡算法，比如这次分配给第 N 个 CTS，下次则分配给第 N+1 个 CTS，当到了最后一个，就从第一个开始；为了更好地实现容错，我采用了当第 N 个 CTS 失效的时候，把它正在处理的业务转到第 N+1 个上面继续处理，这样大大增加了系统的可用性，可以为客户提供更好的服务；此外，我还采用了数据库连接池的技术，大大缩短了数据库处理速度，提高了系统运行速度。

2、并行批量。银行系统每天都要处理大量的数据，为了确保白天的业务能顺利进行，有一部分的帐务处理，比如一部分内部户帐务处理，或者代理收费业务和总帐与分户帐核对等功能就要到晚上批量地去处理，但是，这部分数据在数据集中之后就显得更加庞大，我行以前采用串行提交批量作业的办法，远远不能适应数据中心亿万级的数据处理要求，在与其他技术骨干讨论之后，并经过充分的论证和试验，我决定采用了并行批量的技术，所谓的并行批量，就是在利用 IBM 的 OPC(Tivoli Operations，Planning and Control)技术，把批量作业按时间和业务处理先后顺序由操作员统一提交的基础上，再利用 DB2 的 PARTITION 技术，把几个地区分到一个 PATITON 里面分别处理，大大提高了银行系统的数据处理速度，确保了远期结售汇系统三层结构的先进性。在并行批量的设计过程中，我考虑到批量作业有可能因为网络错误或者资源冲突等原因而中断，这样在编写批量程序和作业的时候必须支持断点重提，以确保生产的顺利进行。

由于我软件三层结构设计得当，并采取了有效的措施去解决设计中遇到的问题，远期结售汇系统最后按照计划完成并顺利投产，不但保证了系统的开发性开放性、可用性和互用性，取得了良好的社会效益和经济效益，而且我的软件三层结构设计得到了同事和领导的一致认同与称赞，为我行以后系统的开发打下了良好的基础。

在总结经验的同时，我也看到了我在软件三层结构设计中的不足之处：首先，负载算法过于简单，容易造成系统的负荷不均衡：由于每个业务的处理时间不一样，有的可能差距很远，简单的顺序加一负载分配算法就容易造成负载不均衡，但是如果专门设置一个分配器，则增加了一次网络通

讯，使得系统的速度变慢，这样对响应速度要求很高的银行系统来说也是不可行的，于是我决定采用基于统计的分配算法，即在收到请求的时候，根据预先设定的权值，按概率，直接分配给 CTS。其次，由于批量作业顺序设计得不过够严谨等各种原因，容易造成资源冲突：在远期结售汇系统运行了一段时间之后，数据中心的维护人员发现了，系统有的时候会出现资源冲突现象，在经过仔细的分析之后，我发现，由于每天各个业务的业务量大小不一样，顺序的两个作业之间访问同一个表的时候便会产生资源冲突，另外，在 OPC 作业运行的过程中，操作员提交的其他作业与这个时间的 OPC 作业产生也有可能产生资源冲突。对于第一种情况，可以在不影响业务的情况下调整作业顺序或者对于查询作业运用 DB2 的共享锁的技术，而第二种情况则要制定规范，规定在某时间断内不允许提交某些作业来解决。为了更好地开展系统分析工作，我将在以后的工作实践中不断地学习，提高自身素质和能力，为我国的软件事业贡献自己的微薄力量。

云原生架构

云原生架构是基于云原生技术的一组架构原则和设计模式的集合，旨在将云应用中的非业务代码部分进行最大化的剥离，从而让云设施接管应用中原有的大量非功能特性(如弹性、韧性、安全、可观测性、灰度等)，使业务不再有非功能性业务中断困扰的同时，具备轻量、敏捷、高度自动化的特点。

云原生的代码通常包括三部分：业务代码、三方软件、处理非功能特性的代码。从业务代码中剥离大量非功能性特性(不会是所有，比如易用性还不能剥离)到 IaaS 和 PaaS 中，从而减少业务代码开发人员的技术关注范围，通过云厂商的专业性提升应用的非功能性能力。

具备云原生架构的应用可以最大程度利用云服务和提升软件交付能力，进一步加快软件开发。其特点包括：代码结构发生巨大变化、非功能性特性大量委托、高度自动化的软件交付。

云原生架构原则

服务化原则：拆分为微服务架构、小服务架构，分别迭代。

弹性原则：系统的部署规模可以随着业务量的变化而自动伸缩，无须根据事先的容量规划准备固定的硬件和软件资源。

可观测原则：通过日志、链路跟踪和度量等手段，使得一次点击背后的多次服务调用的耗时、返回值和参数都清晰可见。

韧性原则：当软件所依赖的软硬件组件出现各种异常时，软件表现出来的抵御能力。

所有过程自动化原则：一方面标准化企业内部的软件交付过程，另一方面在标准化的基础上进行自动化，通过配置数据自描述和面向终态的交付过程，让自动化工具理解交付目标和环境差异，

实现整个软件交付和运维的自动化。

零信任原则：默认情况下不应该信任网络内部和外部的任何人/设备/系统，需要基于认证和授权重构访问控制的信任基础，以身份为中心。

架构持续演进原则：云原生架构本身也必须是一个具备持续演进能力的架构。

主要架构模式

1.服务化架构模式：典型模式是微服务和小服务模式。通过服务化架构，把代码模块关系和部署关系进行分离，每个接口可以部署不同数量的实例，单独扩缩容，从而使得整体的部署更经济。

2.Mesh 化架构模式：把中间件框架(如 RPC、缓存、异步消息等)从业务进程中分离，让中间件 SDK 与业务代码进一步解耦，从而使得中间件升级对业务进程没有影响，甚至迁移到另外一个平台的中间件也对业务透明。分离后在业务进程中只保留很“薄”的 Client 部分，Client 通常很少变化，只负责与 Mesh 进程通信，原来需要在 SDK 中处理的流量控制、安全等逻辑由 Mesh 进程完成。

3.Serverless 模式：将“部署”这个动作从运维中“收走”，使开发者不用关心应用运行地点、操作系统、网络配置、CPU 性能等，从架构抽象上看，当业务流量到来/业务事件发生时，云会启动或调度一个已启动的业务进程进行处理，处理完成后云自动会关闭/调度业务进程，等待下一次触发，也就是把应用的整个运行都委托给云。

4.存储计算分离模式：在云环境中，推荐把各类暂态数据(如 session)、结构化和非结构化持久数据都采用云服务来保存，从而实现存储计算分离。

5.分布式事务模式：大颗粒度的业务需要访问多个微服务，必然带来分布式事务问题，否则数据就会出现不一致。架构师需要根据不同的场景选择合适的分布式事务模式。

6.可观测架构：可观测架构包括 Logging、Tracing、Metrics 三个方面，其中 Logging 提供多个级别的详细信息跟踪，由应用开发者主动提供；Tracing 提供一个请求从前端到后端的完整调用链路跟踪，对于分布式场景尤其有用；Metrics 则提供对系统量化的多维度度量。

7.事件驱动架构：本质上是一种应用/组件间的集成架构模式。可用于服务解耦、增强服务韧性、数据变化通知等场景中。

云原生架构范文

2022 年 5 月，我单位联合某高校研发了《程序在线评测比赛考试系统》。系统以程序代码在线提交自动评测功能为核心，分为题库模块、评测机模块、实验作业模块、考试模块、比赛模块、抄袭判定模块、用户管理模块等，支持对接教务平台。在项目中我担任系统架构师，负责架构设计工作。

本文以该系统为例，主要论述了云原生架构在项目中的具体应用。系统以 Spring Cloud 微服务框架开发，分为前端 Web 服务、平台保障服务、业务服务三部分。前端 Web 服务由负载均衡与服务器集群结合，实现高并发的前台界面；平台保障服务以 Eureka 为中心，由 API 网关、服务注册中心、监控平台等构成，实现基础服务框架；业务服务划分为多个微服务，基于 Docker 容器，协同工作实现具体业务功能。最终系统顺利上线，获得用户一致好评。

笔者在一个专为高校建设计算机专业智能教学一体化平台的单位任职，过往成果有《计算机组成原理仿真实验系统》等。2020 年 5 月，我单位联合某大学研发了《程序在线评测比赛考试系统》项目(以下简称为“OJ 系统”)，以取代原有传统的编程上机考试平台。

系统以程序代码的在线提交自动评测功能为核心，主要分为题库模块、评测机模块、实验作业模块、考试模块、比赛模块、抄袭判定模块、用户管理模块等。题库模块主要负责试题和测试用例的管理，用户根据试题要求编写程序代码提交到系统，系统将测试用例与程序代码发送给评测机模块，由评测机自动编译、执行、判分，并将结果发送给其他相关模块进行统计；实验作业模块用于在线布置作业，从题库中选取试题，设置截止日期等要求；考试模块用于学生在线考试，按教师预先设置的参数自动从题库随机抽题生成试卷，以及向教务平台上传考试成绩；比赛模块主要用于 ACM 竞赛的培训；抄袭判定模块用于鉴定代码与他人代码雷同率；用户管理模块负责用户信息的管理。在这个项目中，我担任了系统架构师的职务，主要负责系统的架构设计相关工作。

云原生架构以微服务和容器技术为代表，有服务化、强韧性、可观测性和自动化四类设计原则。通过服务化的设计原则，应用被分解为多个服务，可分别选择不同的技术，单个服务模块很容易开发、理解和维护，无需协调其他服务对本服务的影响；通过强韧性的设计原则，微服务可以分布式云化部署，负载均衡管理请求的分发，避免单机失败对整体服务的影响，以及弹性调整资源容量；通过可观测性的设计原则，能够对系统进行健康检查、指标监控、日志管理和链路追踪，提高系统运维、管理和排错能力；通过自动化的设计原则，可实现系统的自动化部署、自动化扩展伸缩、自动化运维、持续交付和集成，有效减少人工操作的工作量。

OJ 系统基于 Spring Cloud 微服务框架开发，将平台服务划分为三类，分别为前端 Web 服务、平台保障服务、业务服务。下面针对这三类服务展开具体说明。

1.前端 Web 服务

前端 Web 服务主要提供给用户使用的界面，分为前置 Nginx 负载均衡服务器、前端网站 Nginx 集群。当用户通过网络访问系统时，首先会访问到前置的 Nginx 负载均衡服务器，负载均衡服务器会将请求转发到前端网站的 Nginx 集群，前端网站通过发起 Http 请求来和后端交互，具体是通过 Ajax 方式来调用后端 RESTAPI 接口。用户访问网站通过前置的 Nginx 负载均衡服务器来转发到前端

网站集群，以起到将用户请求进行分流的作用。当前端网站集群中的部分服务发生故障时，系统仍可正常地对外提供服务。前置 Nginx 负载均衡服务器使用软件反向代理的方式来实现负载均衡，部署为路由模式，系统内部网络与外部网络分属于不同的逻辑网络，以实现系统内部与外部网络的隔离。在负载均衡算法的选择上，使用最小连接法，每当用户的请求来临时，任务分发单元会将任务平滑分配给最小连接数的前端网站节点，这样的架构以廉价且透明的方式扩展了服务器和网络的带宽，可以大大提升系统的并发量，同时保证网站前端整体的稳定性和可靠性。

2.平台保障服务

平台保障服务用以实现后端微服务的基础框架，包括 API 路由网关、服务注册中心、服务监控组件。API 网关收到前端的请求，不会直接调用后端的业务服务，而是首先会从服务注册中心根据当前请求来获取对应的服务配置，随后通过服务配置再调用已注册的服务。当后端微服务存在多个实例时，将采取负载均衡的方式调用。服务注册中心是整个云原生架构体系的核心部分，由 Spring Cloud 的 Eureka 组件来实现，专门提供微服务的服务注册和发现功能，涉及三种角色：服务提供者、服务消费者和服务注册中心。API 路由网关、所有业务服务，以及服务监控平台组件都注册到服务注册中心。通过服务注册中心两两互相注册、API 路由网关向服务注册中心注册多个实例等方式，来实现后端整体服务的高可靠性。服务监控平台通过注册到服务注册中心，获取所有注册到服务注册中心的后端业务服务，从而监控到所有后端业务服务的运行状态信息，最后收集并展示整个微服务系统的运行状态，更进一步保证整个后端的服务质量。

3.业务服务

业务服务按功能模块，相应划分为题库服务、评测机服务、考试服务、比赛服务、抄袭判定服务等。各服务单独打包，基于 Docker 容器，连同运行环境一起封装，根据实际情况可在一台或多台物理机同时部署多个实例，服务启动后会将自身信息注册到服务注册中心。服务间协同工作，通过松耦合的服务发现机制，动态调用对方 RESTAPI 接口。对于压力较大的服务，如评测机服务、抄袭判定服务等，将部署为多实例集群。以在线考试功能为例，用户进入考试时，考试服务核验考生信息通过，调用题库服务，题库服务返回试题，由考试服务组合成试卷，返回前端显示。用户交卷时，提交的程序代码到达考试服务，考试服务拆分后分发给题库服务，题库服务将程序代码和测试用例送入 MQ 队列排队。然后由负载均衡机制，依次将队列中待评测程序分发给评测机服务编译、执行、判分，完成评测后，题库服务统计试题通过率，考试服务统计成绩并向前端显示。在此期间服务请求者无需了解其他服务对数据如何具体处理和分析。

系统自 2023 年 2 月正式上线已运行一年有余，在学校的日常教学考试和竞赛培训中投入使用，至今已有 3000 名以上的学生用户，评测了 70000 份以上的程序代码，获得了单位同事领导和学校教

师们的一致好评。在开发和试运行过程中，主要遇到了两个问题。一是跨域问题。OJ 系统前后端分离，前端通过 Ajax 访问后端服务。由于浏览器同源策略的限制，导致前端 UI 无法正常访问不同端口和 IP 的后端服务。我们利用 Spring Boot 后端的 Cors 跨域机制解决了该问题。二是评测机宕机问题。评测机服务需要执行用户提交的代码，部分用户短时间内提交了大量不安全代码，导致评测机集群中所有实例全部宕机。我们引入心跳机制、快照回滚机制，以及基于机器学习技术的预判断机制，使评测机宕机时能够在 10 秒内自动重置恢复运行，最终解决了该问题。

实践证明，OJ 系统项目能够顺利上线，并且稳定运行，与系统采用了合适的架构设计密不可分。经过这次云原生架构的方法和实施的效果后，我也看到了自己身上的不足之处，在未来还会不断更新知识，完善本系统在各方面的设计，使整个 OJ 系统能够更加好用，更有效地服务于高校师生。

3.2 论面向服务架构设计及其应用

诸葛老师预测，2024 下半年微服务考查的可能性较大，建议同学们以微服务为主来写。

试题 论面向服务架构设计及其应用

企业应用集成(Enterprise Application Integration，EAI)是每个企业部必须要面对的实际问题。面向服务的企业应用集成是一种基于面向服务体系结构(Service-Oriented Architecture，SOA)的新型企业应用集成技术，强调将企业和组织内部的资源和业务功能暴露为服务，实现资源共享和系统之间的互操作性，并支持快速地将新的应用以服务的形式加入到已有的集成环境中，增强企业 IT 环境的灵活性。

请围绕“SOA 在企业集成架构设计中的应用”论题，依次从以下 3 个方面进行论述。

1、概要叙述你参与管理和实施的企业应用集成项目及你在其中所担任的主要工作。

2、具体论述 SOA 架构的内容、特点，以及你熟悉的工具和环境对 SOA 的支持，在应用中重点解决了哪些问题。

3、通过你的切身实践详细论述 SOA 在企业应用集成中发挥的作用和优势。

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

SOA 架构的内容、特点，以及你熟悉的工具和环境对 SOA 的支持。

在应用中重点解决了哪些问题。

问题 3 要点：

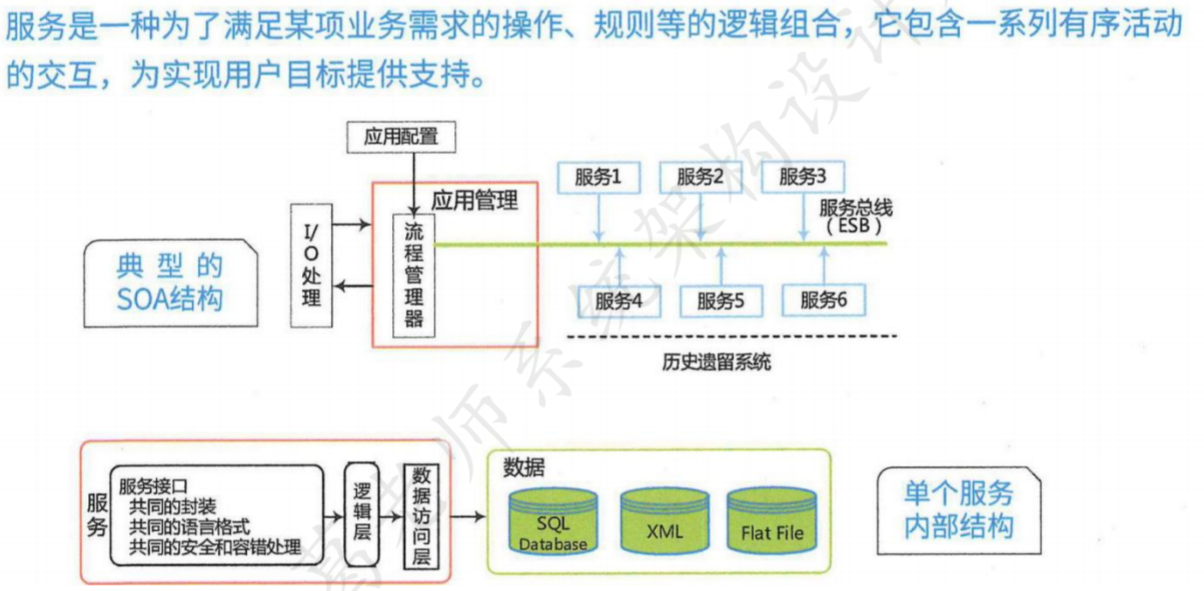
SOA 在企业应用集成中发挥的作用和优势。

理论素材准备

SOA 是一种粗粒度、松耦合服务架构，服务之间通过简单、精确定义接口进行通信，不涉及底层编程接口和通信模型。

在 SOA 中，服务是一种为了满足某项业务需求的操作、规则等的逻辑组合，它包含一系列有序活动的交互，为实现用户目标提供支持。

SOA 并不仅仅是一种开发方法，还具有管理上的优点，管理员可直接管理开发人员所构建的相同服务。多个服务通过企业服务总线提出服务请求，由应用管理来进行处理，如下图所示：



实施 SOA 的关键目标是实现企业 IT 资产重用的最大化，在实施 SOA 过程中要牢记以下特征：可从企业外部访问、随时可用(服务请求能被及时响应)、粗粒度接口(粗粒度提供一项特定的业务功能，而细粒度服务代表了技术构件方法)、服务分级、松散耦合(服务提供者和服务使用者分离)、可重用的服务及服务接口设计管理、标准化的接口(WSDL、SOAP、XML 是核心)、支持各种消息模式、精确定义的服务接口。

从基于对象到基于构件再到基于服务，架构越来越松散耦合，粒度越来越粗，接口越来越标准。

基于服务的构件与传统构件的区别有四点：

①服务构件粗粒度，传统构件细粒度居多；

②服务构件的接口时标准的，主要是 WSDL 接口，而传统构件常以具体 API 形式出现；

③服务构件的实现与语言是无关的，而传统构件常绑定某种特定的语言；

④服务构件可以通过构件容器提供 QoS 的服务，而传统构件完全由程序代码直接控制。

SOA 的实现方式

WebService

服务提供者、服务注册中心(中介，提供交易平台，可有可无)、服务请求者。服务提供者将服务描述发布到服务注册中心，供服务请求者查找，查找到后，服务请求者将绑定查找结果。如下图：

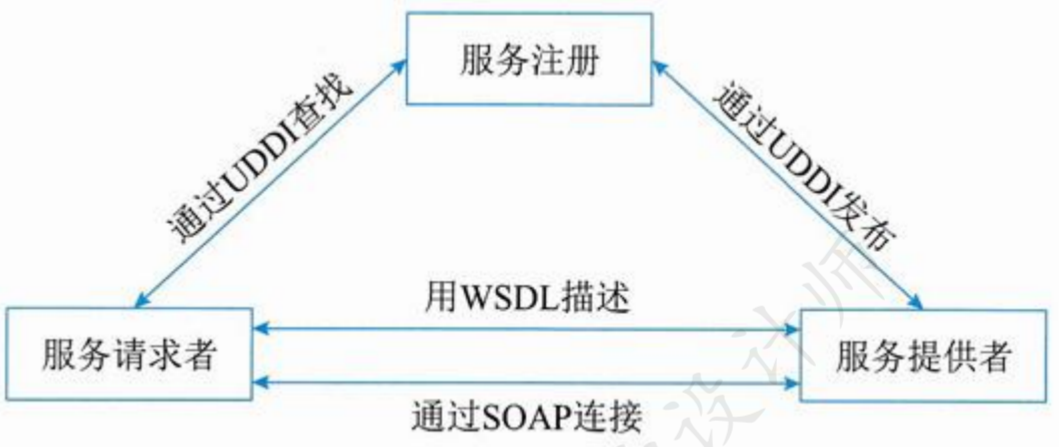


图 15-4 基本 Web 服务协议

合格范文参考

摘要：2021 年 8 月，我参与了胶凝砂砾石坝施工质量监控系统的开发工作，该系统旨在帮助水利工程建设法人单位、施工企业、监理机构及相关政府部门解决水利工程建设施工质量监控和工程项目管理等问题。我在该项目中担任系统架构设计师，主要负责该系统的架构设计工作。本文以胶凝砂砾石坝施工质量监控系统为例，主要论述了 SOA 在企业集成架构设计中的具体应用。服务提供者主要完成服务的设计、描述、定义和发布等相关工作；服务注册中心保证该系统各个模块、服务的相互独立性与松耦合；服务请求者通过 WebService 技术调用服务。实践证明，通过以上技术的应用有效实现了资源共享和系统间的互操作性，提高了系统的灵活性，最终系统顺利上线，获得用户一致好评。

正文：胶凝砂砾石坝是在面板坝和碾压混凝土重力坝基础上发展起来的一种新坝型，其特点是采用胶凝砂砾石材料筑坝，使用高效率的土石方运输机械和压实机械施工。与常规坝型相比，胶凝砂砾石坝在适用性和经济性方面具有独特的优势，可以就地、就近取材，不需设置集料筛分，施工进度快，施工工序简单高效，因而要求施工过程紧凑，高峰期筑坝效率要求高，这给施工质量控制带来了一定的困难和风险，需要综合考虑影响施工质量的各方面因素，尽量采用自动化监控手段，加强实时质量监控力度，这使胶凝砂砾石坝施工质量监控系统应运而生。

2021 年 8 月，我参与了胶凝砂砾石坝施工质量监控系统的开发工作，担任该系统的系统架构师，主要负责该系统的系统分析及设计工作。该系统的主要功能模块包括采料监控、运料监控、拌合监控、碾压监控和温湿度监控等。旨在帮助水利工程建设法人单位、施工企业、监理机构及相关政府部门，解决水利工程建设施工质量监控和工程项目管理等问题，通过信息技术和施工信息现场采集、实时传输、统一存储、科学分析和在线处理，及时生成质量监控报表和发布质量预警信息，提高水利工程建设管理和科学化、现代化和信息化，落实法人负责、监理控制、施工保证、政府监督等各项职能。因此，要满足该系统的需求，选择一种合适的架构技术至关重要。

SOA 是一种应用程序架构，在这种架构中，所有功能都定义为独立的服务，服务之间通过交互和协调完成业务的整体逻辑。SOA 指定了一组实体，包括服务提供者、服务消费者、服务注册表、服务条款、服务代理和服务契约，这些实体详细说明了如何提供和消费服务。服务提供者提供符合契约的服务，并将他们发布到服务代理。这些服务是自我包含的、无状态的实体，可以由多个组件组成。服务代理者作为存储库、目录库或票据交换所，产生由服务提供者发布的事先定义的标准化接口，使得服务可以提供给在任何异构平台和任何用户接口使用。这种松散耦合和跨技术实现，使各服务在交互过程中无需考虑双方的内部实现细节、实现技术、以及部署在什么平台上，服务消费者只需要提出服务请求，就可以发现并调用其他的软件服务得到答案。SOA 作为一种粗粒度、松耦合的架构，具有松散耦合、粗粒度服务、标准化的接口、位置和传输协议透明、服务的封装和重用、服务的互操作等几个特点。

该系统要求开发周期短，系统灵活性高等，结合 SOA 的特点，我们最终采用了面向服务的、基于 SOA 的企业应用集成。下面具体论述其应用过程。

1、服务提供者

服务提供者主要完成服务的设计、描述、定义和发布等相关工作。经过对水利行业施工工程及施工工艺的深入研究，通过查阅《胶凝砂砾石坝施工指南》等相关资料，根据企业应用集成的要求，对胶凝砂砾石坝施工质量监控系统的业务流程进行梳理；综合考虑服务粗粒度、松耦合、自包含和模块化等特点进行服务的设计。为了避免服务通信期间，信息量过大，服务之间交互过于频繁，尽量的减少了服务的数量。同时，为了保证服务自身功能的完整性，尽可能的减少服务与系统之间的通信，在胶凝砂砾石坝施工质量监控系统的分析与开发过程中，先行设计，提取出了两个必要，急需的服务便于日后集成使用，其中包括拌合监控中标准拌合比对比服务和碾压监控中的碾压轨迹生成服务。

在标准拌合比对比服务中主要实现针对现有拌合配比与标准拌合比的对比，以判断现有拌合配比是否符合标准的工作。由于胶凝砂砾石坝就地、就近取材的特性，因此在不同的水利施工工地所

使用的采料也不尽相同，标准拌合比对比服务预留了标准拌合比的输入接口，以适应不同的需求。在碾压轨迹生成服务中主要实现读取定位信息绘制碾压轨迹，以监控是否存在漏碾和欠碾的情况。由于受到胶凝砂砾石坝选址和机密程度的限制，定位信息可以选择 GPS 或者超宽带技术，但是两种定位的方式的数据格式并不相同，因此碾压轨迹生成服务的开发中预留了两种数据格式的接口来读取定位信息。待完成服务设计之后，服务提供者采用 WSDL 进行服务描述，而后再利用 UDDI 技术将这些服务信息发布至服务注册中心，公布查找和定位服务的方法。

2、服务注册中心和服务请求者

在胶凝砂砾石坝施工质量监控系统采用了服务注册中心。服务注册中心比不是一个必选角色，但是为了保证该系统各个模块、服务的相互独立性与松耦合，在该系统中依然保留了服务注册中心。同时，服务注册中心的存在也使得服务请求者与服务提供者之间进一步解耦。在服务注册中心包含有已发布的标准拌合比对比服务与碾压监控中的碾压轨迹生成服务的描述信息，其描述信息主要包括服务功能描述、参数描述、接口定义、信息传递等相关信息。

服务请求者通过 WebService 技术调用服务。当服务完成发布，在服务请求者要使用已发布的服务。利用 WebService 技术在拌合监控阶段，通过服务注册中心获取拌合监控中标准拌合比对比服务的相关功能，接口，参数及其返回值等相关服务信息；之后使用 WebService 技术传递服务所需的标准拌合比等相关参数，进而调用该服务相关的运算、处理和分析。利用 WebService 技术在拌合监控阶段，通过服务注册中心获取实时施工数据采集处理服务的服务定义和功能，接口，参数及其返回值等相关服务信息；之后根据施工工地的具体情况选择不同的定位方式，传递服务所需相关参数，最后实时施工数据采集处理服务运行结束返回的绘制碾压轨迹坐标点同样是利用 WebService 技术传递至服务请求者。服务请求者接收到碾压轨迹的坐标点后最终完成碾压轨迹的绘制工作并在界面中将其呈现出来。在这期间服务请求者无需了解服务是如何对数据进行处理和分析的。

整个项目历时 10 个月开发，于 2022 年 6 月完成交付，到目前运行稳定。通过在水利施工工地等恶略环境下的一段时间的使用，用户普遍反馈良好。总体来讲，选用 SOA 有如下优势：1、系统更易维护。当需求发生变化时，不需要修改提供业务服务接口，只需要调整业务服务流程或者修改操作即可。2、更高的可用性。该特点是在于服务提供者和服务请求者的松散耦合关系上得以发挥与体现。这种没有绑定在特定实现上、具有中立的接口定义的特征称为服务之间的松耦合。松耦合有两个明显的优势，一是它的灵活性，其独立于实现服务的硬件平台、操作系统和编程语言；二是当组成整个应用程序的每个服务的内部结构和实现逐渐地发生改变时，它能够继续存在。3、更好的伸缩性，依靠业务服务设计、开发和部署等所采用的架构模型实现伸缩性。使得服务提供者可以互相彼此独立地进行调整，以满足新的服务需求。

实践证明，SOA 技术的使用大大提高了系统开发效率，节省了开发和维护成本，使得系统具有更好的开放性、易扩展性和可维护性，从项目完工后的使用效果来看，达到了预期目的。

微服务架构范文

2022 年 8 月，我单位联合 xxx 有限公司开发了省 xxx 综合应用管理平台，作为公司核心技术骨干，我担任了系统架构师的职务，主要负责 xxx 应用系统架构体系设计及核心组件的开发工作。该系统按照省机关业务类型划分，依次包含基础功能支撑板块、平台资源管理板块、煤炭能源板块、油气板块、新能源能源板块、电力板块、安全监管板块、经济运行板块、智能数据分析业务以及数据可视化板块，业务范围依次涵盖省煤炭、电力、油气、新能源等能源领域。

本论文首先介绍构建 xxx 应用系统的项目背景，然后分析了微服务技术架构对于该项目的必要性，并以能源云应用系统作为例，结合微服务架构的特性与实际情况，分别讨论了微服务技术架构的应用情况。实践证明，在大型的应用系统构建过程中，使用微服务技术架构，能够实现各应用分区自治、庞大业务的有效管理及业务功能灵活拓展的优势。

xxx 综合应用管理平台，是机关响应国家“十四五”规划所采取的数字信息化措施的开创性项目，旨在深化运用国家以及省市政务信息资源，加强政务信息共享，实现数据编目、数据整合、能源应用服务，规范政务信息资源社会化增值开发利用工作，合理规划政务信息的采集(煤炭、油气、新能源、电力等领域)，加强政务信息资源管理。项目的总体目标为：理清能源数据家底，形成能源数据资源目录；实现省能源数据的统一综合管控；基于能源大数据，支撑能源全产业的决策与分析；通过省级数据共享与开放平台向兄弟单位共享能源数据、向社会企业与公众开放脱敏数据。

在项目早期，我们组织了相关承建企业及核心用户，一起进行了项目需求的评审，拟在确定项目的研发计划、细化项目需求，从而进一步确定采取的系统软件架构。项目整体涉及了能源领域的众多业务，体系繁杂且比较庞大，部分业务有着相似的数据支撑组件而部分业务之间又不存在过多的信息交互，基于此，我提出了项目整体采取微服务架构体系构建的方案，并陈述了按照业务板块划分、基础业务拆分方式规划微服务的必要性。在专家评审过程中，通过以往采用微服务架构体系规划项目的经验，最终确定了该系统架构设计方案。

微服务架构体系作为目前 IT 领域主流的技术，有服务化、强韧性、可观测性和自动化四类设计原则。通过服务化的设计原则，应用被分解为多个服务，可分别选择不同的技术，单个服务模块很容易开发、理解和维护，无需协调其他服务对本服务的影响；通过强韧性的设计原则，微服务可以分布式云化部署，负载均衡管理请求的分发，避免单机失败对整体服务的影响，以及弹性调整资源容量；通过可观测性的设计原则，能够对系统进行健康检查、指标监控、日志管理和链路追踪，提

高系统运维、管理和排错能力；通过自动化的设计原则，可实现系统的自动化部署、自动化扩展伸缩、自动化运维、持续交付和集成，有效减少人工操作的工作量。

系统开发过程中，主要采取 Spring Cloud Alibaba 技术架构作为微服务架构的实现方案，系统采取 Jenkins+Docker 一体化部署方式实现微服务的部署，前端采用 Vue3.0 开发架构，通过 Nginx 实现 HTTP 请求的动态负载及业务服务的调用层次抽象管理。采用该体系具备众多优势，下面就其特点、开发过程及系统上线后的实际情况进行结合，从而说明本项目采取此方案的好处、遇到的问题及其解决方案。

1、以微服务独立部署，实现各应用的独自管理，却又可以简便的进行交互。

每个服务都是一个独立的项目，可以独立部署，不依赖于其他服务，耦合性低。在系统构建过程中，我结合项目整体需求将应用系统拆分为了众多微服务，按照业务领域、使用用户类型对象进行划分，包含以提供基础数据支撑的平台服务，主要提供用户管理、能源企业管理、企业部门管理及业务数据字典管理等服务；以提供能源领域服务的四大板块服务，包括煤炭、油气、新能源、电力；以提供综合数据应用管理的上层众多汇总型服务。按照这种方式划分，由三个承建企业依次划分职责，同步开发，大大的提高了项目整体完成的效率。若服务之间需要进行通信，只需基于微服务体系的消息交互标准，以 Rest 标准化接口通过 FeignAPI 组件实现远程服务调用，通过 Nacos 构建的统一网关，实现方便快捷的交互。

2、服务的快速启动

合理拆分之后服务由于依赖的库及代码量的减少，能够极大的提高服务的启动速度。虽然合理拆分能够提高系统启动速度，但也增加了系统服务的数量。基于此，我通过采取构建统一的打包平台方案，选取了 Jenkins+Docker 镜像结合的解决方案，通过 Docker 将每一个微服务打包为至少一个能独立运行的容器，并通过 Docker-Compose 描述这些镜像服务的关系，使用 Jenkins 进行脚本的统一集成，所有服务均以可视化方式展现在 Jenkins 平台，很好的解决了服务数量增多之后的管理问题。

3、职责专一，由专门的承建企业负责专门的工作职能

本项目涉及领域众多、周期长，服务的拆分有利于团队之间的分工。在项目开发计划中，我们将服务划分过后，由不同的企业负责不同的服务，例如我司承建四大能源领域板块的构建等，在项目开发过程中，除了特殊的业务流程需要服务之间进行信息交互外，大部分情况下均无需在意其他团队的开发进度。

4、服务可以动态按需扩容

当某个服务的访问量较大时，我们只需要简单的增加服务的申请资源或者增加服务实例数量，即可零成本的实现服务扩容。在应用部署的早期，我们将所有的服务实例均部署为 3 个，通过 Nginx

实现一级均衡的同时，也采取了微服务的 Ribbon 负载体系。例如对于煤炭服务的访问，可动态负载到该 3 个不同的煤炭服务实例，该 3 个部署实例位于不同的服务器上，动态负载的同时也保障了服务不会出现单点故障问题。但随着系统用户(各省市机关、各企业用户)的加入及业务数据日益累计，整个系统出现了一定的性能问题。经过排查分析，发现煤炭企业用户会在每天下午 2 点左右集中上报生产数据，导致同一时刻大量的报表填报请求，导致出现了性能瓶颈，需要进行扩容，通过修改 Jenkins 配置文件，增加打包镜像数量将原有的 3 个报表服务实例扩容一倍解决了问题。

5、服务的复用

每个服务都提供 RESTAPI，所有的基础服务都按照尽可能高的内聚度进行抽取。类似于组件开发方法，可将一些底层的服务进行归纳总结，方便应用到以后的项目中，提高企业的生产效率。在本项目中，众多基础服务大部分均复用可以往团队开发的组件。譬如报表动态生成服务，该服务是以往能源产业几乎一致的需求服务，可通过该服务动态配置填报报表对象、填报周期、时间截止等。

经过我和团队的不懈努力，历时一年，项目终于于 XXXX 年 XX 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，通过此类较大应用的开发使得公司的应用规划能力得以提升。但是，在实施过程中，也暴露了一些具体问题，例如微服务之间接口交互时，由于业务复杂，简单的消息调用无法满足繁忙场景，当交互频率增大到一个数量级时需要建立具有动态优先级调整机制的处理队列等等，这些问题通过引入消息队列组件 Kafka 得以妥善解决，没有影响到项目的运行情况。也在项目的实施过程中，发现了自己的一些短板，例如在安全性处理方面，还需要从服务部署层面、三方安全软件集成方面提升，由于自己从事的几乎都是政府职能项目，安全性是尤为需要考虑的质量属性，今后也会在这方面深入研究，力求使自己的架构实施能力更加全面、稳固，为国家政府信息化建设规划贡献自己的一份力。

3.3 论软件的系统测试及应用

2024 上半年架构师考到了“论单元测试及运用”，因此，测试可能也是重要的考点了。预测 2024 下半年可能会考查测试相关内容，测试包括单元测试、集成测试、系统测试、静态测试、动态测试等，这些知识点都需要掌握。

试题 论软件的系统测试及其应用

软件测试是软件交付客户前必须要完成的重要步骤之一，目前仍是发现软件错误(缺陷)的主要手段。系统测试是将已经确认的软件、计算机硬件、外设、网络等其他元素结合在一起，针对整个系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地

方，从而提出更加完善的方案。系统测试的主要内容包括功能性测试、健壮性测试、性能测试、用户界面测试、安全性测试、安装与反安装测试等。

请围绕“软件的系统测试及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及你在其中所担任的主要工作。

2.详细论述软件的系统测试的主要活动及其所包含的主要内容，并说明功能性测试和性能测试的主要的目的。

3.结合你具体参与管理和开发的实际项目，概要叙述如何采用软件的系统测试方法进行系统测试，说明具体实施过程以及应用效果。

找准核心论点

问题 1 要点

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

软件的系统测试的主要活动及其所包含的主要内容。

功能性测试和性能测试的主要的目的。

问题 3 要点：

采用软件的系统测试方法进行系统测试的具体实施过程以及应用效果。

还有哪些地方值得改进或提高。

理论素材准备

系统测试的对象是完整的、集成的计算机系统，系统测试的目的是在真实系统工作环境下，验证完整的软件配置项能否和系统正确连接，并满足系统/子系统设计文档和软件开发合同规定的要求。系统测试的技术依据是用户需求或开发合同，除应满足一般测试的准入条件外，在进行系统测试前，还应确认被测系统的所有配置项已通过测试，对需要固化运行的软件还应提供固件。

一般来说，系统测试的主要内容包括功能测试、健壮性测试、性能测试、用户界面测试、安全性测试、安装与反安装测试等，其中，最重要的工作是进行功能测试与性能测试。功能测试主要采用黑盒测试方法；性能测试主要验证软件系统在承担一定负载的情况下所表现出来的特性是否符合客户的需要，主要指标有响应时间、吞吐量、并发用户数和资源利用率等。

功能测试的目的很简单，测试系统是否达到了用户明确提出的需求及隐含需求。

性能测试的目的是验证软件系统是否能够达到用户提出的性能指标，同时发现软件系统中存在的性能瓶颈，并优化软件，最后起到优化系统的目的。具体来说，包括以下四个方面：

（1）发现缺陷。软件的某些缺陷与软件性能密切相关，针对这些缺陷的测试一般需要伴随着性能测试进行。

（2）性能调优。与调试不同，性能调优并不一定针对发现的性能缺陷，也可能是为了更好地发挥系统的潜能。

（3）评估系统的能力。软件性能测试不仅需要测试软件在规定条件下是否满足性能需求，往往还需要测试能够满足性能需求的条件极限。

（4）验证稳定性和可靠性：在一定负载下测试一定的时间，是评估系统稳定性和可靠性是否满足要求的唯一方法。

合格范文参考

摘要：2023 年 6 月，我所在公司组织了某市环境影响评价会商系统一期的开发工作。我有幸作为该项目的技术负责人参与整个开发过程。该项目主要业务需求是技术评估流程业务，包括规划环评流程、建设项目流程、竣工验收项目流程、环保专项资金评估流程。本文以该系统为例，主要论述了软件系统测试技术在该项目中的具体应用。在系统测试阶段，我们从功能测试和用户界面测试以及性能测试三个方面对该项目进行了测试工作。功能测试主要验证业务流程的正确以及表单数据的合法输入；用户界面测试主要验证界面和原型的匹配以及浏览器的兼容性；性能测试主要验证系统最大在线人数的并发。通过以上技术使得项目的测试工作顺利进行，最终项目成功上线，获得用户一致好评。

正文：为促进某市环评信息化的建设，某市环境工程评估中心向市环境工程评估中心递交了《某市环境工程评估中心关于申请环评基础数据库建设试点单位的请示》，环保部评估中心已正式回函，同意将该市作为全国环评基础数据库的试点单位，并与环保部评估中心开展国家级环评数据库的技术合作和共建共享工作。我所在的公司是一家专注环保业务的软件开发公司，其中环境影响评价是我们的主要业务之一。通过招标，我们公司成功的中标了这个项目，2023 年 6 月，我们正式进行项目的开发工作，2024 年 3 月，完成了项目的验收工作。我作为技术负责人全程参与了这个项目的开发、测试工作。

环境影响评价基础数据库是指支撑环境影响评价全生命周期，以及环境影响评价过程产生的数据的集合。主要包括支撑数据、业务数据、管理数据三大库群。环评基础数据库的建设过程实际上是构建“横向”支撑数据库群和“纵向”业务数据库群，以及搭建管理数据库群的过程。“横向”支撑数据库群主要包括支撑环境影响评价全生命周期(环评、技术评估、审批管理、监督后评价、公众参与等)的数据资源。“纵向”业务数据库群包括环境影响评价全生命周期产生的所有数据资源，涉及战略环

评、规划环评、区域环评、项目环评等，管理数据库群主要包括为环评管理服务的基础支撑数据，如环评资质管理数据、环评从业人员数据、技术评估专家库等。该系统在功能和性能方面要求较高，因此，采用合理的系统测试方法显得至关重要。

软件测试是软件交付客户前必须要完成的重要步骤之一，目前仍是发现软件错误(缺陷)的主要手段。系统测试是将已经确认的软件、计算机硬件、外设、网络等其他元素结合在一起，针对整个系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地方，从而提出更加完善的方案。系统测试的主要内容包括功能性测试、健壮性测试、性能测试、用户界面测试、安全性测试、安装与反安装测试等。功能性测试主要通过白盒的测试方法，主要目的是验证系统是否达到了用户提出的需求或者隐性的需求。用户界面测试主要的目的是验证系统的界面设计是否达到客户的要求，浏览器的兼容性。性能测试主要是系统在一定负载的情况下表现出来的性能是否达到客户的性能指标，同时发现系统中的性能瓶颈、并优化软件最终达到优化系统的目的。结合我们系统的实际情况，我们对系统进行了功能性测试、用户界面测试和性能测试。

一、功能性测试

该阶段的主要任务是通过黑盒测试的方式验证系统是否符合需求规格说明书上的业务。在实际测试中，首先测试人员根据需求规格说明书，制定了测试计划和录入了测试用例，然后在按模块的一个一个进行测试。在测试中主要对业务流程是否符合需求和流程表单的各个节点的表单的必填项以及代办测试。由于这个系统的流程比较多，流程节点也比较多，所以测试的过程中需要频繁的切换账号来验证流程的正确性。比如项目负责人录入了一个项目，提交后到部门主任审核并分配办理人员。测试人员需要先用项目负责人登陆系统，然后录入一个项目并提交，退出登陆后再用部门主任账号登陆，查看是否有代办事宜，通过代办进入审核页面，然后提交下一步时候选择业务办理人员，这样流程就流转到业务办理人员的代办了，业务办理人员登陆后，也是通过代办进入后在这个阶段需要验证表单的一些字段是必填的是否验证了必填。测试人员通过这样的测试流程一个一个流程和模块的测试，最终完成了功能性测试的目的，发现了一些 bug 并提交到了 QC 系统。

二、用户界面测试

该阶段的主要目的测试系统的用户界面是否符合用户的要求，用户的要求是界面大方简洁、兼容目前市场的主流浏览器并且系统的代办业务在手机端也可以兼容显示。对于界面要求，测试人员主要是根据美工提供的页面原型和实际的系统对比，测试效果是否和页面原型的效果一致。对于浏览器的兼容测试，测试人员使用的方法是下载并安装目前的主流浏览器，使用每个浏览器访问业务系统，验证在各个浏览器中系统是否有不兼容的情况，比如界面变形、有些验证失效、表格对齐等。对于代办业务在手机端显示，测试人员通过在手机登陆系统后，查看代办列表是否有提醒提示，通

过提示是否能够看到代办的项目名称和提交的时间。通过这个阶段的测试，测试人员发现了在一些功能的问题，比如在建设项目审批流程的项目负责人编写报告书阶段，其中附件报告书是必须上传的，但是现在不上传也能提交。在界面测试的时候发现表格的样式和美工提供的原型相差很大，不支持只适应，手机代办页面，代办列表不能自适应手机的界面。通过这个阶段的测试发现了一些界面上的问题也提交到了 QC 系统。

三、性能测试

该阶段的主要任务是系统在一定负载的情况下表现出来的性能是否达到客户的性能指标，该系统的主要的性能指标是单台服务器在 500 人同时在线的情况下系统是否能提供正常的服务。为了完成这个测试，测试人员要求开发人员单独在一台配置为 4 核 16G 内存硬盘为机械硬盘的服务器上部署该系统，然后通过性能测试软件 Load runner 对系统进行性能测试。测试的过程为：并发的人数以 10、30、50、100、200、300、400、500 的方式依次进行登陆和退出的测试，并对 TPS、响应时间、点击率、数据库 CPU 负载、应用服务器 CPU 负载和服务器吞吐量进行了记录。在测试的过程中当用户并发达到 400 的时候，系统的性能明显的下降了很多，通过向开发人员反映，经过几次的优化后，最终在并发为 500 的时候，系统的性能没有明显的下降，达到了客户提出的性能的要求。

2024 年 3 月，系统顺利通过验收并且上线运行。系统上线后，系统功能和性能都达到用户的要求，得到各个部门领导和业务人员的一致好评。但是系统上线一段时间以后也出现了一些不足的地方，比如建设项目流程，在办公室人员核对节点，审核不通过的时候需要上传说明文档，由于测试人员的疏忽没有测试到这个要求，导则在实际流程过程中项目审核不通过的时候也可以提交，还要系统上线后很快就发现了问题，我们也及时的修复了这个漏洞。我们准备在二期的项目中对测试这块加大要求，以保证项目在线上环境尽可能少的出现低级的错误。

实践证明，有效利用多种测试方法充分进行系统测试，可以有效降低项目风险，对项目顺利进行起到至关重要的作用。通过该项目的顺利实施和验收，让我在系统测试方面受益良多，也深刻认识到我们技术工作者要不断学习，拼搏进取，提高自身的素质和能力，为国家的环保事业奉献自己。

3.4 论软件设计方法

2024 下半年，诸葛老师预测会考查软件设计，可能涉及到面向对象的设计(如设计原则和设计模式)、软件设计四个过程(架构设计、接口设计、过程设计、数据设计)、业务流程设计等。

软件设计相关范文较少，下面提供一个性能设计的范文供参考，包含了业务流程优化。

摘要：笔者于 2022 年 5 月参与了某地级市市级机关的电子政务信息系统的建设工作，该电子政

务系统分为三个模块，分别是政府办公自动化模块、政企信息查询模块、公共信息发布模块，笔者在该项目中担任系统架构设计师一职，主要负责系统的架构设计、优化和项目的日常管理工作。在该系统的开发过程当中，由于用户对该系统的性能提出了较高的要求，因此我们在实际的开发过程中，对系统进行了优化设计，具体从三个方面着手：一、软件的业务流程设计，主要针对电子政务信息系统的业务流转方式，对业务流程进行优化设计，压缩和裁减冗余流程，提高作业流程的流转速度；二、数据组织的优化设计，通过采用高性能 DBMS，优化数据库基本表结构，对常检索的表和字段建立索引等方式来优化数据组织，三、软件的结构组织优化，通过高度模块化、采用大量成熟构件的手段来进行优化。该系统上线运行后，整体表现出色，基本实现了用户需求，但也存在一些尚待解决的问题。

正文：笔者于 2022 年 5 月参与了某地级市市级机关的电子政务信息系统的建设工作，该电子政务系统是当地政府的数字化政府工程的关键部分，该政务信息系统主要分为三个子模块，分别是政府办公自动化模块、政企信息查询模块、公共信息发布模块，笔者在该项目中担任系统架构设计师，主要负责系统的架构设计、优化和项目的日常管理工作。

在对该市级机关原有的政务信息系统进行调研时，笔者发现该机关原有的政务信息系统存在种种性能上的不足，主要表现在：一、业务流程繁琐，单项业务的流转速度较慢，业务分支过于复杂；二、数据库版本陈旧，性能较差，数据的读写操作速度较慢；三、软件系统本身的运行速度较慢，很多时候其响应时间甚至超过 10 秒的基线时间，并且经常出现因兼容性问题而导致的系统出错甚至崩溃。综上所述，由于上述问题的存在，该机关信息化委员会在制定新的电子政务系统开发规划方案时，将业务流程的优化、数据结构组织的优化、软件结构的优化三点列为新版电子政务系统的三项主要性能指标。

经过反复的讨论，项目组最终决定采用对业务流程进行优化设计、压缩和裁减冗余流程、提高作业流程的流转速度等方法来优化电子政务系统的业务流程；通过采用 ORACLE 数据库等高性能数据管理系统、优化数据库基本表结构、对常检索的表盒字段建立索引等方式来优化数据组织结构；通过采用高度模块化的设计方式进行软件模块设计，采用大量成熟构件的开发手段来对系统的组织结构进行优化。

在对业务流程的优化过程中，我们对业务流程进行了规划、分解、组合，尽可能将业务流程梳理清楚，每一条业务流程的中间环节原则上不超过 5~7 个，并且每一个业务流程所完成的工作尽可能清减，尽量避免在一项业务流程中完成过多的业务工作，造成业务流程的运行速度变慢，对冗余的业务流程进行裁减、压缩，将多余、不必要的业务流程删除，将可以合并的业务流程进行合并和整合，使其整合为一个流程，减少在系统中流转的业务模块，提高系统的运作速度，加快业务的流

转。经过上述的规划与设计，有效地提高了业务的流转速度，根据后期的测试和运行结果，在处理同一项业务流程的时候，新系统比旧系统的速度提高了 25~30%。

在对数据组织结构进行优化的过程中，我们主要采用了三种方式对数据组织结构进行优化：1、采用 ORACLE 大型数据库管理系统替换原有的 ACCESS 数据库，ORACLE 数据库拥有出色的性能、完善的厂商服务支持、强大的扩展性和丰富的支持接口，尤其是 ORACLE 数据库具有优秀的网络环境支持能力，这是 ACCESS 数据库所不能比拟的，由于该机关的电子政务系统运行在网络环境中，对网络的要求较高，因此 ORACLE 数据库有效地提高了该系统的性能。2、对数据库的基本表结构进行优化，在对新版政务信息系统的基本表结构进行设计的时候，以 3NF 的表结构为主，并且采用垂直分片的数据分割方式，将容量较大的几个主表分布存放在不同的数据库服务器上，以减轻查询负担，加快查询速度，特别需要说明的是，对于一些查询量较大，而更新频率相对较小的数据表，我们采用了基于 2NF 的设计方式，适当提高数据的冗余，以加快数据表的查询速度。3、对常检索的数据表和字段，建立了索引以加快检索的速度，为了避免因索引过多反而导致性能下降的情况出现，我们仅在查询量较大、更新频率较低的表上建立索引，并且索引字段选择为主键或常用字段。

在对软件结构进行优化的设计过程当中，我们主要采用了模块化设计方法和构件化设计方法来提高软件结构的性能。其中模块化设计方法主要是通过将系统的功能组织为一个个高度内聚、低度耦合、大粒度行为的模块，每个模块完成的工作尽可能单一，尽可能杜绝一个模块完成多项业务工作的情况，模块与模块之间的联系被降低到最低程度，彼此之间的相互通信由集中式控制模块进行统一的调度与控制。构件化设计方法主要指在设计和开发的过程中，我们大量采用性能成熟的构件进行开发，构件的获取方式主要有从现有的构件库中检索和提取、外购商业化构建、自行开发等，通过大量采用构件进行开发，有效地提高了该系统的性能和可靠性，并提高了该系统的扩展性。

在系统的开发工作完成之后，我们对该系统开展了测试工作，测试工作主要围绕三个方面进行，其中业务流程的测试主要测试系统流程的顺畅程度，一项业务流程的处理时间能否达到理想的要求；数据读写操作的测试主要测试数据的大批量读取和写入的操作速度，和响应时间是否达到性能要求，系统在高负荷运行的状态下，数据的读写速度是否让人可以接受，以及系统的负载均衡能力等；软件结构的测试主要是测试该系统在实际运行时的速度、健壮性等指标是否达标，系统的业务处理速度和响应速度是否令人满意等。测试工作结束后，我们将测试结果编制为测试结果报告书，并上报该机关的信息化建设委员会，经该委员会的审核，该报告书最终得以通过。

该系统上线后，运行情况出色，经用户反映，有效地提高了系统性能，基本满足了用户的需求。在项目总结大会上，我们对项目进行了总结，会议认为，该项目整体上是成功的，但是也存在着一些问题，主要有以下两点：一是在对业务流程进行优化和压缩时，有的业务部门基于自己部门的利

益，对业务清减和压缩工作有所抵制，二是在采用第三方厂商所提供的构件进行构件化开发时，由于第三方厂商所提供的构件为通用化构件，给个性化定制工作带来了困难。对于第一个问题，笔者认为应当由该机关的高层领导牵头，召开相关的工作会议，在会议上对业务的清减、压缩、改造等牵扯到部门业务整合改造的工作作出统一的部署，对于第二个问题，笔者认为在购买第三方厂商所开发的商业化构件时，应当采购可扩展性较好，售后服务支持较完善的构件产品。

以下是设计和需求结合范文，并用到了面向对象设计

摘要：2022 年 5 月，我参加了某市供电公司《电力营销管理信息系统》的开发工作，并担任系统架构师一职，主要负责系统分析和架构设计。该系统分为：业扩管理、计量管理、电量电费核算管理、收费与账户管理、线损管理等五个模块。采用了 Struts+Spring+Hibernate 的主流 Web 框架，降低了开发的难度和成本，降低了模块之间的耦合度，提高软件的可维护性和可扩展性。需求分析是软件开发过程中最为重要的环节，是项目成败的关键。软件需求包括：功能需求、非功能需求、设计约束三个方面。本文以该项目为例，结合作者的实践，论述了系统设计中对用户需求的把握。我们采用了面向对象的分析方法，从需求捕获、需求分析、需求规格化、需求验证 4 个环节来论述。还介绍了需求分析中使用 UML 的类图、用例图、状态图等，最后介绍了存在的问题及改进措施。

正文：2022 年 5 月，我参加了某市供电公司《电力营销管理信息系统》的开发工作，并担任系统架构师一职，主要负责系统分析和架构设计。该供电公司年供电量在 10 亿度以上，计量点 915 个，固定大客户 209 个。以前的业务流程是：电话包装、手工派单、自主开发的 VFP 系统算费、财务系统收费开票。业务之间不连续，无法有效对业务的整个过程进行管理，客户满意度低。随着业务量的扩大，供电公司认识到存在的不足，急需一套电力营销管理信息系统来解决上述问题。以系统的建设促进用电管理水平的提升，以电力信息化推动电力企业现代化。杜绝重复投资，总体规划，实现用电管理信息的高速交汇和决策，提升用户的满意度，降低管理的成本。

该系统包括：业扩管理、计量管理、电量电费核算管理、收费与账户管理、线损管理五个模块。采用了面向服务 SOA 架构模型，Struts+Spring+Hibernate 的主流 Web 框架，开发工具采用 MyEclipse10.0。硬件方面，两台 IBMX3650 服务器安装 Oracle10g 做数据库服务器，在两台服务器上搭建了数据库高级复制功能，实现了数据自动同步。两台 IBMX3650 以双机热备的方式做营销应用服务器，两台服务器上运行集群软件，通过“心跳”检测对方的状态，发现故障能动切换。一台 IBMX3650 做算费服务器。

需求分析是软件开发过程中最重要的环节，是项目成败的关键。软件需求就是系统必须完成的事及必须具备的品质。具体来说，软件需求包括：功能需求、非功能需求、设计约束三个方面。功

能需求是系统必须完成的事，即向用户提供有用的功能，产品必须执行的动作。非功能需求是产品必须拥有的属性和品质，如可靠性、性能、响应时间、安全性等。设计约束是限制条件，如用户必须运行在 UNIX 系统下，必须使用的数据库管理系统等。需求分析是一项技术含量很高的工作。包括需求捕获、需求分析、需求规格化、需求验证 4 个环节。需求捕获是收集需求信息；需求分析是把收集的信息进行分析、建立模型；需求规格化是编制软件需求规格说明书；最后客户和管理层进行验证。上述 4 个步骤不是瀑布式，是迭代的演化过程。

在项目的开发中，我们采用了统一建模语言(UML)，并使用了 Rational Rose 工具。在建模中使用了 UML 的类图、用例图、部署图和活动图。

1、需求捕获。我们制定捕获计划时，在计划中我们建立了一张表格，左边填写想要了解的内容，右边填写预计信息的来源，这样建立起一一对应的关系，使我们的捕获有的放矢、更高效。首先，我们通过对关键人员进行访谈，了解用户现在存在的问题和继续解决的问题。对原有的系统进行分析(如自主开发的 VFP 算费系统、财务系统等)，找出那些可以满足用户的需要，那些不能满足继续改进。财务系统开出的发票是有统一的标准，我们只要把开票所需的数据提供给财务系统就可以满足用户的要求。自主开发的算费系统就不能满足对电费计算的多样化需求。查阅历史报表，对关键业务进行现场观摩、参与业务流程。经过上述过程对用户的需求有了大概的了解，经过整理后邀请客户代表、系统分析人员、开发团队代表等召开联合讨论会，讨论那些需求被正确理解，那些没有考虑，那些还需要修改。迭代上述过程，直到把计划中想了解的问题全部的到确定。

2、需求分析。我们采用面向对象的分析方法进行求分析。需要标示于系统相关的对象，并描述对象的属性和它们之间的关系，还要了解这些对象是如何协同完成系统功能。首先要寻找类。找出需求描述中的名词和名词短语，从中提取对象与属性。找出动词和动词短语，从中提取操作于关联。在完成了类型寻找之后，我们用 UML 的类图工具对找到的类记录，确定类之间的层次、关系等。然后建立系统的用例模型。用例是系统执行的一系列动作，产生特定参与者可见的价值结果。先识别系统的参与者，是同系统交互的所有事物。不仅指人，还有其他的系统、硬件设备。参与者一定在系统之外，我们可以通过谁使用这个系统，谁维护这个系统，谁为这个系统提供信息，谁从这个系统获取信息等方面来识别参与者。在本系统中参与者包括：系统业务人员、管理人员、集抄系统、财务系统、抄表机等。找到参与者之后，接下来是仔细检查参与者，为每个参与者确定用例。主要依据已经获得的特征表。利用 UML 的用例图工具对参与者和用例进行描述。获得了用例模型的框架，接下来要细化用例描述，包括：用例名称、简要说明、事件流、前置条件、后置条件、扩展点、优先级等。用协作图描述了协作实现特定功能的对象。用状态图描述了对象在生命周期的转换情况。

3、需求规格化。用需求分析的结果编写软件需求规格说明书。需求规格说明书是软件开发过程

中最为重要的文档之一，要保证完整性、正确性、可行性。为了让非技术人员方便阅读和理解，尽量通过自然语言和简单图表来描述，防止造成不必要的误会。我们对传统的用例说明书进行了一些简单的调整，加入了基于用例分析的图形元素，对参与者和用例采用了客户容易的词语来描述，如业务人员、业务经理、财务人员、计算电费等。

4、需求验证。由用户关键代表、领域专家、我们公司的领导组成的验证小组对需求规格说明书进行验证评审，最终通过了评审。结束语：该项目于 2023 年 10 月通过了客户的验收正式运行，大大提高了供电公司的用电管理水平，提升了客户的满意度，降低了管理的成本。项目的成功源于各方面共同努力的结果。需求分析是软件开发过程中最重要的过程，是项目成败的关键。正确理解客户的要求是需求分析的基础，这是一个复杂多变的过程，需要分析人员运用各种技巧，按照需求捕获、需求分析、需求规格化、需求验证来迭代演化的过程。

这次需求分析我也遇到了一些问题，主要是维护 Word 文档与模板的一致性。我们使用 Rational Rose 建模，使用 Word 编写文档，通过截图的方式把模型插入文档，一旦模型发生变化还要重新插入到文档中，比较繁琐容易出错。在以后的系统分析中，我打算使用 Rational 公司的 Soda 工具，自动的把 Rose 模型插入到 Word 文档中。

3.5 论文软件开发模型及应用

2024 下半年，诸葛老师预测可能会考到一个关于开发方法或开发模型的主题，如统一过程。同学们可以重点写这个，课程也会进一步讲解，明确统一过程的九大工作流、四个阶段和三大特点。

试题　论软件开发模型及应用

软件开发模型(Software Development Model)是指软件开发全部过程、活动和任务的结构框架。软件开发过程包括需求、设计、编码和测试等阶段，有时也包括维护阶段。软件开发模型能清晰、直观地表达软件开发全过程，明确规定了要完成的主要任务和活动，用来作为软件项目工作的基础。对于不同的软件项目，针对应用需求、项目复杂程度、规模等不同要求，可以采用不同的开发模型，并采用相应的人员组织策略、管理方法、工具和环境。

请围绕“软件开发模型及应用”论题，依次从以下三个方面进行论述。

1.简要叙述你参与的软件开发项目以及你所承担的主要工作。

2.列举出几种典型的软件开发模型，并概要论述每种软件开发模型的主要思想和技术特点。

3.根据你所参与的项目中使用的软件开发模型，具体阐述使用方法和实施效果。

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

常用的开发模型：瀑布模型、快速原型模型、演化模型、增量模型、螺旋模型、喷泉模型、敏捷方法。

每种软件开发模型的主要思想和技术特点。

问题 3 要点：

在项目中所采用的开发模型，具体实施效果如何。

还有哪些地方值得改进或提高。

理论素材准备

常见开发模型及特点

瀑布模型：严格遵循软件生命周期各阶段的固定顺序，一个阶段完成再进入另一个阶段。其优点是可以使过程比较规范化，有利于评审；缺点在于过于理想，缺乏灵活性，容易产生需求偏差。

快速原型模型：对于许多需求不够明确的项目，比较适合采用该模型。它采用了一种动态定义需求的方法，通过快速地建立一个能够反映用户主要需求的软件原型，让用户在计算机上使用它，了解其概要，再根据反馈的结果进行修改，因此能够充分体现用户的参与和决策。

演化模型：也是一种原型化开发，但与快速原型不同的是，快速原型模型在获得真实需求时，就将抛弃原型。而演化模型则不然，它将从初始的模型中逐渐演化为最终软件产品，是一种“渐进式”原型法。

增量模型：它采用的是一种“递增式”模型，它将软件产品划分成为一系列的增量构件，分别进行设计、编码、集成和测试。

螺旋模型：结合了瀑布模型和演化模型的优点，最主要的特点在于加入了风险分析。它是由制定计划、风险分析、实施工程、客户评估这一循环组成的，它最初从概念项目开始第一个螺旋。

喷泉模型：主要用于描述面向对象的开发过程，最核心的特点是迭代。所有的开发活动没有明显的边界，允许各种开发活动交叉进行。

统一过程(UP)：统一过程是一个通用过程框架，可以用于种类广泛的软件系统、不同的应用领域、不同的组织类型、不同的性能水平和不同的项目规模。UP 是基于构件的，在为软件系统建模时，UP 使用的是 UML。与其他软件过程相比，UP 具有三个显著的特点，即用例驱动、以架构为中心、迭代和增量。

敏捷方法：敏捷方法是一种以人为核心、迭代、循序渐进的开发方法。在敏捷方法中，软件项目的构建被切分成多个子项目，各个子项目成果都经过测试，具备集成和可运行的特征。在敏捷方法中，从开发者的角度来看，主要的关注点有短平快的会议、小版本发布、较少的文档、合作为重、客户直接参与、自动化测试、适应性计划调整和结对编程；从管理者的角度来看，主要的关注点有测试驱动开发、持续集成和重构。

合格范文参考

摘要：2022 年 10 月，我参加了 X 市公安局数据中心支撑平台项目的开发，该项目主要目的是开发一个通用性的框架平台，其主要功能是提供一个统一、高效和具有强大扩展能力的警务数据支撑平台，包括一体化公安数据处理平台、可再生的公安数据服务支撑平台、开放式的公安应用平台、健全的安全与运维监控平台，并将该市现有的各种警务信息系统遗产进行通用化封装和集成到该数据支撑平台上。本文以该项目建设为例，讨论了软件开发模型及其应用的问题，重点论述了根据项目特点和实际情况选择开发模型以及应用统一过程进行系统开发的过程。我们确定使用 RUP 统一过程来实施项目开发，分 3 个阶段进了行 4 次迭代完成了项目开发任务。我在项目开发中担任系统架构设计师，主要负责系统架构设计工作。

正文：2022 年 10 月，我所在的公司通过公开招标竞标的方式获得了 X 市公安局数据中心支撑平台建设项目，工期 240 天。公司组建了由 11 人组成的项目开发团队，我担任系统架构设计师，主要负责系统架构设计工作。

X 市公安局已开展了十多年的信息化建设工作，取得了相当的成果和积累了许多信息化经验。随着警务信息化的不断推进和发展，现阶段公安机关视频、卡口、人像、案情文本等非结构化数据呈现几何指数增长并凸显其重要性，传统的警务系统已经开始出现疲态，技术瓶颈逐步显现，相关情报研判和案件分析的响应速度越来越慢，甚至有些应用场景已经完全不能支撑。为实现公安信息化“深化建设”和“深度应用”，在公安部、省公安厅的统一部署下，X 市公安局决定尽快实施数据中心支撑平台项目，建设一体化公安数据处理平台、可再生的公安数据服务支撑平台、开放式的公安应用平台、健全的安全与运维监控平台，充分发挥现有资源作用和新一代信息技术优势，形成具有公安特色、符合公安业务需求的数据支撑平台。

一、选择开发模型

软件开发模型是软件开发全过程、活动和任务的框架，是软件系统开发的重要基础。在软件工程发展历程中，出现了线性开发模型(如瀑布模型)和迭代开发模型(如螺旋模型、统一过程和敏捷开发模型等)，其中瀑布开发模型是按照“问题定义-需求分析-系统设计-系统开发-测试与运行维护”的

流程实施软件系统的开发，该模型是以需求明确为前提的，其主要缺点是无法适应需求的变化以及缺乏用户参与。

在 X 市公安局数据中心支撑平台项目开发过程中，我们确定要使用迭代的模型来开发各个子系统，但可供选择的开发模型有敏捷开发方法和 RUP 统一开发过程等。敏捷开发方法强调“个体和交互胜过过程和工具、可工作的软件胜过大量的文档、客户合作胜过合同谈判、响应变化胜过遵循计划”，而统一过程是“以架构为中心、用例驱动”的模型，二者都强调以用户为核心，主要的区别在于敏捷开发是一种轻量级的迭代开发模型，统一过程是一种重量级的迭代模型。基于以下的因素，我们最终确定使用统一过程来开发系统。

1、敏捷方法和 RUP 方法在对待风险态度上有明显的区别，敏捷方法在项目后期也接受需求和技术架构的变更，而 RUP 方法强调在项目早期消除主要的风险，以保证项目开发的进度和质量。由于本项目是政府安全机关主导的信息化建设，具有相当程度上的严肃性和敏感性，容不得任何闪失，需要将各种风险降到最低程度。

2、敏捷开发方法针对中小型软件系统开发具有较好的效果，但随着项目规模的不断增大，迭代次数会陡增，给项目管理和实施带来极大困难。

3、项目团队在敏捷开发方面积累的经验相对较少，学习成本较高，项目进度和质量不易把控。

二、统一过程开发

统一过程一般分为初始阶段、细化阶段、构建阶段和交付阶段，每执行一遍这四个阶段便完成了一次迭代，是否进行下一次迭代取决于评审目标是否完成。

1、初始阶段

X 市公安局数据中心支撑平台涉及到刑警、经警、交警、户籍警、政府管理部门以及社会公众、团体等众多实体。在初始阶段，首先识别系统的参与者和关键用例，识别出诸如数据标准化、案卷调阅、轨迹跟踪、信息预警、信息布控、视频取证、异地资源互访、分级报警以及卡口数据分析等用例，我们用 Rational Rose 对关键用例进行建模。

其次，根据项目开发背景、要求和特点，我们识别、分析和评价了项目的风险，由于该项目是由政府安全机关主导的信息化建设工程具有一定的严肃性和敏感性，存在的风险不仅仅在于商业和技术风险，更重要的是要保证按时和高质量交付产品，因此，最大的风险是由于该项目本身所具有的复杂性以及人员、进度、成本和质量管理不完善造成进度延迟和质量得不到保证的问题。

最后，该项目不仅需要将 X 市公安局现有的各种系统整合起来，而且还要开发一些新的应用，但由于前期建设的各种系统的数据标准不统一，难以为后续应用提供可靠服务。基于此，我们将 X 市公安局数据中心支撑平台建设总体方案规划为三大阶段：第一阶段为基础平台建设阶段，第二阶

段为平台完善与应用阶段，第三阶段为大规模应用阶段。

2、细化阶段

针对当前识别出的各种项目风险，对其进行分析和评价，鉴于项目工期短、任务重，为保证按时高质量地完成项目开发，项目组经过与公司管理层协商，从其他项目组抽调 2 名经验丰富的业务骨干充实到本项目团队中，以因应项目开发的紧迫性。

X 市公安局数据中心支撑平台需要与公安部、省公安厅实现无缝对接，并且还要整合利用现有系统，我们确定应用 SOA 架构来实施整个项目的开发，利用 WebService 将 X 市公安局现有的数据平台、报警系统、户籍系统等平台封装成标准服务，并利用 ESB 松散耦合起来实现整体业务逻辑。为了按计划实施项目开发，我们首先针对基础数据平台进行开发，建立统一的数据标准化体系，为后续应用系统开发奠定基础，实现以数据推动应用，以数据驱动业务，以数据创新思路。其次，细化关键用例、建立支持环境并将公司现有、可利用的构件挑选出来以备复用。最后，对本阶段工作进行了技术评审。

3、构建阶段

在第一次迭代的构建阶段，我们主要进行数据、服务与管理标准建设、数据标准维护系统开发、数据标准管理系统开发以及数据信息资源库开发等工作，拟定的开发周期为 30 天。

为保证项目开发的总体进度，我们利用甘特图和 PERT 图进行项目进度规划和管理，在时间紧、任务重的状况下，我们利用 WBS 确定了本次迭代的工作范围，并将项目团队分为 4 个工作小组进行并行开发，其中包括一个数据标准建设小组、一个应用系统开发小组、一个信息资源库开发小组以及一个测试小组。4 个小组协调工作，特别值得一提的是，我们在项目开发前期就实施了测试计划制定、测试用例设计并针对系统需求进行了需求测试、功能测试和性能测试等一系列测试工作，避免了报废和返工，保障了项目开发的质量和效率。

4、交付阶段

在交付阶段，我们将前期开发的应用系统移植到 X 市公安局信息中心进行了 Beta 测试，通过测试后将这些应用系统封装成服务“挂载”到服务总线(ESB)上，形成一个可交付的产品版本。同时，我们邀请用户代表、系统分析师、设计与开发小组以及测试人员一起进行技术评审，在确定满足相关功能要求、性能指标并达成共识的情况下，结束本阶段开发。

三、总结

在 X 市公安局数据中心支撑平台项目开发过程中，我们一共进行了 4 次迭代，在每次迭代过程的初始、细化、构建和交付阶段均开展了相应的技术评审，并采取相应的措施保证系统的质量，如期完成了项目开发。同时，我们也遇到了一些问题，如在需求分析师对网络登录时间统计分析不够

细致，导致在后期验收时与用户的要求有偏差，我们及时与用户进行了充分沟通达成先进行验收，并承诺在维护阶段改进该功能，最终得到用户的理解与认可。

3.6 论企业应用集成

系统架构设计有个很重要的作用就是解决信息孤岛问题，因为架构集成考论文也很多，分别会从不同的维度去考，2023 年考到的就是多数据源，其实本质就是数据库的集成。因此诸葛老师预测 2024 下半年可能会考查关于架构集成的内容，可能涉及企业的应用集成、企业集成平台、企业集成架构等。

试题 论企业应用集成

企业应用集成(Enterprise Application Integration，EAI)是完成在组织内、外的各种异构系统，应用和数据源之间共享和交换信息和协作的途径，方法学，标准和技术。企业应用集成所连接的应用包括各种电子商务系统，企业资源规划系统，客户关系管理系统，供应链管理系统，办公自动化系统，数据库系统，数据仓库等。

请围绕“企业应用集成”论题，分别从以下几个方面进行论述：

1、简要叙述你参与的企业应用集成项目(项目的背景、发起单位、目的、项目特点等)。

2、简要叙述企业应用集成的四个层次(方法)?

3、详细论述你参与的项目是采用了哪个层次的集成，如何实施的，效果如何?

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

企业应用集成的四个层次。

问题 3 要点：

采用了哪个层次的集成。

具体实施过程以及应用效果。

理论素材准备

1. 表示集成

表示集成也称为界面集成，这是比较原始和最浅层次的集成，但又是常用的集成。这种方法把

用户界面作为公共的集成点，把原有零散的系统界面集中在一个新的界面中。表示集成是黑盒集成，无需了解程序与数据库的内部构造。常用的集成技术主要有屏幕截取和输入模拟技术。表示集成通常应用于以下几种情况：

(1)在现有的基于终端的应用系统上配置基于 PC 的用户界面。

(2)为用户提供一个看上去统一，但是由多个系统组成的应用系统。

(3)当只有可能在显示界面上实现集成时。表示集成的实现是很简单的，也是很不彻底的，只是做了一层“外装修”，而额外多出来的集成界面也将可能成为系统的性能瓶颈。

2. 数据集成

为了完成控制集成和业务流程集成，必须首先解决数据和数据库的集成问题。在集成之前，必须首先对数据进行标识并编成目录，另外还要确定元数据模型，保证数据在数据库系统中分布和共享。因此，数据集成是白盒集成。有很多不同的中间件工具可以用于数据集成。例如，批量文件传输，即以特定的或是预定的方式在原有系统和新开发的应用系统之间进行文件传输；用于访问不同类型数据库系统的 ODBC 标准接口；向分布式数据库提供连接的数据库访问中间件技术等。通常在以下情况下，

将会使用数据集成：

(1)需要对多种信息源产生的数据进行综合分析和决策。

(2)要处理一些多个应用程序需要访问的公用信息库。

(3)当需要从某数据源获得数据来更新另一个数据源时，特别是它们之间的数据格式不相同时。

相对而言，数据集成比表示集成要更加灵活。但是，当业务逻辑经常发生变化时，数据集成就会面临困难。

3. 控制集成

控制集成也称为功能集成或应用集成，是在业务逻辑层上对应用系统进行集成的。控制集成的集成点存于程序代码中，集成处可能只需简单使用公开的 API 就可以访问，当然也可能需要添加附加的代码来实现。控制集成是黑盒集成。

实现控制集成时，可以借助于远程过程调用或远程方法调用、面向消息的中间件、分布式对象技术和事务处理监控器来实现。控制集成与表示集成、数据集成相比，灵活性更高。表示集成和数据集成适用的环境下，都适用于控制集成。但是，由于控制集成是在业务逻辑层进行的，其复杂度更高一些。而且，很多系统的业务逻辑部分并没有提供 API，这样，集成难度就会更大。

4. 业务流程集成

业务流程集成也称为过程集成，这种集成超越了数据和系统，它由一系列基于标准的、统一数

据格式的工作流组成。当进行业务流程集成时，企业必须对各种业务信息的交换进行定义、授权和管理，以便改进操作、减少成本、提高响应速度。业务流程集成不仅要提供底层应用支撑系统之间的互连，同时要实现存在于企业内部的应用之间，本企业和其他合作伙伴之间的端到端的业务流程的管理，它包括应用集成、B2B 集成、自动化业务流程管理、人工流程管理、企业门户，以及对所有应用系统和流程的管理和监控等。

合格范文参考

摘要：2021 年 10 月，我国某省移动通信有限公司决定启动 VerisBilling6.0 项目，该项目实现了在线计费、离线计费、内容计费、账务处理、信控管理等子系统的整合，我作为系统架构设计师全程参与了项目的建设。本文以 VerisBilling6.0 项目为例，论述企业应用集成中界面集成、数据集成、应用集成在软件集成过程中的实际应用及效果。通过界面集成，实现了账务前台、产品前台、信控前台界面的整合，把几个独立系统经过集成也一个整体展现给用户。通过数据集成，将各之前各系统产生的“信息孤岛”进行了整合，数据的一致性得到有效保障。通过应用集成，从业务逻辑上为各功能系统提供统一的数据接口，使业务逻辑更规范。通过以上集成技术的应用，项目于 2022 年 8 月成功上线，各项性能指标达到客户要求，获得省移动通信公司各级领导的好评。

正文：近几年来某省移动用户增长至 3000 多万，随着移动数据流量资费的新一轮下调，导致 GPRS 数据流量成爆发式增长，OpenBillingNG 版系统在话单处理上瓶颈显现。21 年春节期间，GPRS 日话单达到 30 亿条，话单处理处于积压状态，直到节后两周才将积压话单追完，大量跨月的话单引发了大批用户投诉，给移动业务支撑中心带来的压力非常大；该省移动通信公司相关领导联合系统运营商遂展开会议讨论解决方案，最终决定将该省 OpenBillingNG 版升级至 VerisBilling6.0 版本，以解决 OpenBillingNG 版本遇到的瓶颈问题。作为移动通信 BOSS 业务支撑的核心，VerisBilling6.0 需支持 24×7 连续运行，满足话单的实时处理，还需要把在线计费、离线计费、内容计费、账务处理、产品管理等在 OpenBillingNG 版时独立的系统进行整合。我作为系统架构设计师全程参与了 VerisBilling6.0 项目的建设，VerisBilling6.0 项目由产品管理组、研发组、测试组、对账组、运维组、数据组、专家组共 120 人组成的项目团队，耗时 11 个月完成，项目从 2021 年 10 月初启动，至 2022 年 8 月 30 日上线。

作为系统架构师，我深知在 VerisBilling6.0 项目中系统集成方法对该项目的重要性。通常情况下，企业应用集成中常用的集成有界面集成、数据集成、应用集成等方法。其中界面集成通过将各系统界面进行整合，从而实现了为用户提供一个统一的系统界面的效果，增强了各系统间的交互性能。数据集成的集成点在数据访问层，通过中间件更新数据库的方式，保持数据一致；通过对数据库中

通常需要同步的数据表的集成，实现各系统的数据高效同步，保证了系统数据的一致性。应用集成的集成点都在程序的内部结构中，需要根据业务的实际情况，重组结构，重新开发；是基于业务逻辑层面的集成方法，通过对系统业务逻辑进行集成，为各系统提供统一的业务接口，属于较高层次是集成方式，也是难度较大的集成。三种集成方法相辅相成，互为补充。

如何为 VerisBilling6.0 项目选择合适的集成方法呢?首先，作为 BOSS 系统的核心，VerisBilling6.0 项目是一个庞大、复杂的项目，涉及账务系统前台、产品管理系统前台、信控管理系统前台的界面集成。其次，VerisBilling6.0 项目还涉及计费产品库、账务产品库、信控系统数据库等数据库的整合。最后，项目还涉及在线计费、离线计费、内容计费等几个系统业务逻辑方面的集成。接下来我将从界面集成、数据集成、应用集成三个方面来具体阐述，在 VerisBilling6.0 项目中，我的团队是如何使用这些方式实现企业应用集成的。

界面集成的应用。在 VerisBilling6.0 项目中，我们实现了账务系统前台、产品管理前台、信控管理系统前台进行整合。首先，我们对信控系统页面重新做了布局，将账务系统前台、产品管理前台的功能页合并到信控管理系统的主界面上，并增加了相应的链接功能菜单。其次，在做界面集成时，我们充分的考虑了系统界面的用户友好性，对几个界面做了扁平化设计，并将几个系统的界面样式风格调整一致，使得集成后的系统看上去更像一个整体。最后，我们增加了系统导航功能菜单，完善了系统间的交互性能。经过界面集成，三个系统界面被集成到一个系统页面上，形成了一个整体。当用户登录信控管理系统后台就可以对账务管理系统、产品管理系统、信控管理系统进行操作。由于三个系统都存在独立的系统表，且都实现了自身的权限管理等功能，要想实现单点登录后进行各系统功能的操作，还需要对用户数据进行集成。

数据集成的应用。在 OpenBillingNG 版本中，存在账务产品库、计费产品库、局数据产品库、信控数据库等，在以往的生产过程中，经常会因数据变动后同步不及时而产生错误引发用户投诉。VerisBilling6.0 项目要求将所有产品库进行整合，合并到一个中心数据库，首先，我们对各功能系统的数据库进行分析，将各库中的数据表进行梳理比对，并将平时需要做同步处理的表提取出来分析。接下来，将提取出来的数据表合并到公共数据库，实现了将分散的数据信息进整理合并。最后，将那些系统间无直接关系的表直接割接到公共数据库，经过梳理加工后，完成了对数据的集成，实现了数据的统一管理，数据的一致性得到了保障。经过数据集成，可以有效避免“信息孤岛”的产生，由于系统之间直接调用公共数据表的数据，使得系统数据在任何时候都是完整的、一致的，有效避免数据同步不及时的问题。

应用集成的应用。VerisBilling6.0 项目中，大量的外围系统需要通过接口访问计费 MDB 和账务 MDB。由于这个版本中，MDB 数据表的变动非常大，为了保障 CRM 等外围系统对 MDB 的访问不

受影响，首先，我们对所有的 MDB 接口进行了梳理并同外围系统研发人员进行了确认。接下来，针对每个接口，我们进行了重新定义，并将设计文档发与给外围系统研发负责人，然后按设计文档要求开发出相应的接口。最后，在系统测试过程中，要求各外围系统参与联调测试。由于接口的修改涉及到系统业务逻辑的调整，应用集成难度极大，对外围系统的影响面较广，在该集成方法的使用过程中，参与的人员最多，联调周期最长。应用集成对于各方参与研发的人员综合素质要求也比较高，需要充分考虑逻辑业务的变化对系统的性能的影响，对任何一个接口的疏忽都会产生较为严重的后果。

通过以上集成技术的应用，VerisBilling6.0 项目于 2022 年 8 月底上线，经过半年的运行，系统各项性能指标达到可以要求，并通过客户验收，获得省移动通信公司各级领导好评。在项目结束后的讨论会上，大家也指出了项目中存在一些不足。在项目中由于 MDB 发生了变动，所以需要做业务逻辑的调整，我们在项目中要求所有外围系统都需要参与联调测试，但有几个接口 CRM 没有按要求进行相应的调整，导致系统上线当天出现 CRM 访问 MDB 出现异常。

通过 VerisBilling6.0 项目，使我深刻的认识到在项目实施过程中，每一个细节都需要把控好。首先，在项目中需让专家团队及时对风险进行评估，并针对每个风险点需要提供相应的应对措施，这些措施在项目出现问题时能得到有效的处置。其次，需要制定好实施计划，并严格安计划进行实施，特别是一些需要做联调测试的内容上，必须严格按要求完成，避免出现联调测试不完整这类似问题。最后，在项目中一定要严格把控好每个细节，并实现将系统每个细节的把控落实到个人。

3.7 论软件系统架构风格

试题 论软件系统架构风格

系统架构风格(System Architecture Style)是描述某一特定应用领域中系统组织方式的惯用模式.架构风格定义了一个词汇表和一组约束，词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的口软件系统架构风格反映了领域中众多软件系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。软件系统架构风格的共有部分可以使得不同系统共享同一个实现代码，系统能够按照常用的、规范化的方式来组织，便于不同设计者很容易地理解系统架构。

请以“软件系统架构风格”论题，依次从以下三个方面进行论述：

1.概要叙述你参与分析和开发的软件系统开发项目以及你所担任的主要工作。

2.分析软件系统开发中常用的软件系统架构风格有哪些?详细阐述每种风格的具体含义。

3.详细说明在你所参与的软件系统开发项目中，采用了哪种软件系统架构风格，具体实施效果如何。

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

常用的软件系统架构风格。

每种风格的具体含义。

问题 3 要点：

采用的哪种软件系统架构风格。

具体实施效果如何。

理论素材准备

架构设计的一个核心问题是能否达到架构级的软件复用；

架构风格定义了用于描述系统的术语表和一组指导构建系统的规则；

架构风格反映了领域中众多系统所共有的结构和语义特性，并指导如何将各个构件有效地组织成一个完整的系统；

数据流风格：批处理序列、管道-过滤器；

调用/返回风格：主程序/子程序、面向对象、层次结构；

独立构件风格：进程通信、事件驱动系统(隐式调用)；

虚拟机风格：解释器、基于规则的系统；

仓库风格：数据库系统、超文本系统、黑板系统。

合格范文参考

摘要：本人于 2021 年 2 月参与江苏省某市公交集团“公交车联网一体化“项目，该系统为新能源营运车辆补贴监管、安全监控等方面提供全方位的软件支撑，在该项目组中我担任系统架构师岗位，主要负责整体架构设计与中间件选型。

本文以该车联网项目为例，主要讨论了软件架构风格在该项目中的具体应用。底层架构风格我们采用了虚拟机风格中的解释器，因该公交共有几十种不同的数据协议，使用解释器风格可以满足整车数据协议兼容性需求；中间层关于应用层的数据流转我们采用了独立构件风格中的隐式调用，

这种风格主要用于减低系统间耦合度、简化软件架构，提高可修改性方面的架构属性；应用系统层我们采用了 B/S 的架构风格，统一解决公交行业性难题“实施推广难、维护难”问题。最终项目成功上线，获得用户一致好评。

随着国家十四五计划中-能源战略的深入和推广，该市公交集团自 2021 年 2 月起全面停止采购燃油机公交车，规划到 2024 年纯电公交车采购占比必须在 70%上，同时配套将车联网方面的系统建设被列为工作重点。不管从新能源营运车辆补贴监管、安全监控或者公交公司自身的营运和机护需求，都要求有新的车联网系统对他们进行全方位的支持，而我司是该公交的主要仪表与 can 模块产品的主要供应商，全市 4000 多台车中有 3000 多辆是我司的产品，我司不仅掌握熟悉该公交整车数据而且在车联网底层 can 数据有非常明显的领域知识优势，因此 2021 年 2 月我司被该市公交集团委托建设公交集团车联网一体化项目。本项目组全体成员共有 27 人(不含业主方)，我在项目中为担任系统架构师职务，架构小组共 4 人，我主要职责负责整体架构设计与中间件选型，4 月份完成架构工作，整个项目共耗时了 9 个月，2021 年 11 月顺利通过验收。

在架构工作开始阶段，我们便意识到，架构风格是一组设计原则，是能够提供抽象框架模式，可以为我们的项目提供通用解决方案的，这种能够极大提高软件设计的重用的方法加快我们的建设进程，因此在我司总工程师的建议下，我们使用了虚拟机风格、独立构件风格以及 B/S 架构风格这三种较常用风格。虚拟机风格中的解释器架构风格能够提供灵活的解析引擎，这类风格非常适用于复杂流程的处理。独立构件风格包括进程通讯风格与隐式调用风格，我们为了简化架构复杂度采用了隐式调用风格，通过消息订阅和发布控制系统间信息交互，不仅能减低系统耦合度，而且还提高架构的可修改性。B/S 架构风格是基于浏览器和服务器的软件架构，它主要使用 http 协议进行通信和交互，简化客户端的工作，最终减低了系统推广和维护的难度，以下正文将重点描述架构风格的实施过程和效果。

底层架构我们使用解释器风格来满足整车数据协议兼容性需求。解释器风格是虚拟机风格中的一种，具备良好的灵活性，在本项目中我们的架构设计需要兼容好 86 种不同 can 数据协议，一般来说这种软件编写难度非常高，代码维护难度压力也很大，因此这个解释器的设计任务便很明确了，软件设计需要高度抽象、协议的适配由配置文件来承担。具体的做法如下，我们对各个车厂的 can 数据结构进行了高度抽象，由于 can 数据由很多数据帧组成，每个数据帧容量固定并且标识和数据有明确规定，因此我们将 can 协议中的 ID 和数据进行关系建模，将整体协议标识做为一个根节点，以 canid 作为根节点下的叶子节点，使用 XML 的数据结构映射成了有整车协议链-数据帧-数据字节-数据位这 4 层的数据结构，核心的代码采用 jdom.jar 与 java 的反射机制动态生成 java 对象，搭建一套可以基于可变模板的解释器，协议模板的产生可以由公交公司提供的 excel 协议文档进行转换得到，

解释器支持协议模板热部署，这种可以将透传二进制数据直接映射成 java 的可序列化对象，将数据协议的复杂度简化，后期数据协议更改不会对软件产生影响，仅仅更改协议模板文件即可，最终我们使用了 86 个协议描述文件便兼容了这些复杂的 can 数据协议，规避了 can 数据巨大差异带来的技术风险。

中间层我们使用独立构件风格中的隐式调用来简化构件间的交互复杂度，降低系统耦合度。主要的实现手段是，我们采用了一个开源的消息中间件作为连接构件，这个构件是 Apache 基金会下的核心开源项目 Activemq，它是一款消息服务器，其性能和稳定性久经考验。由上文提到的解释器解析出对象化数据经过 Activemq 分发到各个订阅此消息的应用系统，这些应用系统包括运营指挥调度、自动化机护、新能源电池安全监控等，这种多 Web 应用的情况非常适合采用消息发布与消息订阅的机制，能够有效解决耦合问题，我们在编码的过程中发现只要采用这种风格的 Web 应用，整个迭代过程效率极高，错误率降低，而且我们使用的 Spring 框架，消息队列的管理完全基于配置，清晰简单，维护性良好，例如整车安全主题、运营调度主题、机护维修主题等消息队列分类清晰，可以随时修改其结构也能够随时增其他主题的消息队列，不同的 Web 系统监听的队列也可以随时变换组合，基于消息中间件的架构设计能够让系统的构件化思路得到良好实施，总体来说这种架构风格带来了非常清晰的数据流转架构，简化了编码难度，减低本项目的二次开发的难度。

应用系统层我们主要采用 B/S 的架构风格，主要用于解决公交推广难、维护难得问题。公交行业有一个明显的特点，公交子公司分布在全市各个地区，路途很远，且都是内网通讯，车联网络也是走的 APN 专网，一般是无法远程支持的，这给我们的系统推广以及后期维护带来了很大难题，我们可以想象如果使用 C/S 架构，更新客户端一旦遇到问题很可能需要全市各个站点跑一遍。这让我们在系统推广和维护方面面临较大压力。我们采用的 B/S 架构风格能够解决这个难题，并充分考量可现在的相关技术成熟度，例如现在的 Html5 完全能够实现以前客户端的功能，项目中我们使用了大量的前端缓存技术与 Websocket 技术，能够满足公交用户实时性交互等需求。这种风格中页面和逻辑处理存储在 Web 服务器上，维护和软件升级只要更新服务器端即可，及时生效，用户体验较好，例如界面上需要优化，改一下 Javascript 脚本或者 CSS 文件就可以马上看到效果了。

项目于 2021 年 11 月完成验收，这 1 年内共经历了 2 次大批量新购公交车辆接入，这几次接入过程平稳顺利，其中协议解释器软件性能没有出现过问题，消息中间件的性能经过多次调优吞吐量也接近了硬盘 IO 极限，满足当前的消息交互总量，另外由于我们的项目多次紧急状态下能够快速适应 can 协议变动，得到过业主的邮件表扬。除了业主机房几次突发性的网络故障外，项目至今还未有重大的生产事故，项目组现在留 1 个开发人员和 1 个售后在维护，系统的维护量是可控的，系统运行也比较稳定。

不足之处有两个方面，第一在架构设计的过程中我们忽略 PC 配置，个别 PC 因为需要兼容老的应用软件不允许系统升级，这些电脑系统老旧，其浏览器不支持 Html5，导致了系统推广障碍。第二在系统容灾方面还有待改善。针对第一种问题，我们通过技术研讨会说服可业主新购 PC，采用两台机器同时使用方式解决。针对第二种问题我方采用了服务器冗余和心跳监测等策略，在一台服务暂停的情况下，另外一台服务接管，以增加可用性。

3.8 论高可靠性系统中软件容错技术的应用

2023 下半年架构师论文考到了可靠性的分析和评价，结合近两年可靠性开始在选择题占比增加，这可能意味着可靠性慢慢会变得重要，因此诸葛老师预测 2024 下半年可能会考查可靠性设计。

试题 论高可靠性系统中软件容错技术的应用

容错技术是当前计算机领域研究的热点之一，是提高整个系统可靠性的有效途径，许多重要行业(如航空、航天、电力、银行等)对计算机系统提出了高可靠、高可用、高安全的要求，用于保障系统的连续工作，当硬件或软件发生故障后，计算机系统能快速完成故障的定位与处理，确保系统正常工作。

对于可靠性要求高的系统，在系统设计中应充分考虑系统的容错能力，通常，在硬件配置上，采用了冗余备份的方法，以便在资源上保证系统的可靠性。在软件设计上，主要考虑对错误(故障)的过滤、定位和处理，软件的容错算法是软件系统需要解决的关键技术，也是充分发挥硬件资源效率，提高系统可靠性的关键。

请围绕“高可靠性系统中软件容错技术的应用”论题，依次从以下三个方面进行论述。

1.简述你参与设计和开发的、与容错相关的软件项目以及你所承担的主要工作。

2.具体论述你在设计软件时，如何考虑容错问题，采用了哪几种容错技术和方法。

3.分析你所采用的容错方法是否达到系统的可靠性和实时性要求。

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

如何考虑容错问题。

采用的哪几种容错技术和方法。

问题 3 要点：

采用的容错方法是否达到系统的可靠性和实时性要求。

理论素材准备

系统可靠性是系统在规定的时间内及规定的环境条件下，完成规定功能的能力，也就是系统无故障运行的概率。

系统可用性是指在某个给定时间点上系统能够按照需求执行的概率。

可靠度就是系统在规定的条件下、规定的时间内不发生失效的概率。

失效率又称风险函数，也可以称为条件失效强度，是指运行至此刻系统未出现失效的情况下，单位时间系统出现失效的概率。

冗余技术

提高系统可靠性的技术可以分为避错(排错)技术和容错技术。避错是通过技术评审、系统测试和正确性证明等技术，在系统正式运行之前避免、发现和改正错误。

容错是指系统在运行过程中发生一定的硬件故障或软件错误时，仍能保持正常工作而不影响正确结果的一种性能或措施。容错技术主要是采用冗余方法来消除故障的影响。

冗余是指在正常系统运行所需的基础上加上一定数量的资源，包括信息、时间、硬件和软件。冗余是容错技术的基础，通过冗余资源的加入，可以使系统的可靠性得到较大的提高。主要的冗余技术有结构冗余(硬件冗余和软件冗余)、信息冗余、时间冗余和冗余附加 4 种。

软件容错的主要方法是提供足够的冗余信息和算法程序，使系统在实际运行时能够及时发现程序设计错误，采取补救措施，以提高系统可靠性，保证整个系统的正常运行。软件容错技术主要有 N 版本程序设计、恢复块方法和防卫式程序设计等。

N 版本程序设计：是一种静态的故障屏蔽技术，其设计思想是用 N 个具有相同功能的程序同时执行一项计算，结果通过多数表决来选择。其中 N 个版本的程序必须由不同的人独立设计，使用不同的方法、设计语言、开发环境和工具来实现，目的是减少 N 个版本的程序在表决点上相关错误的概率。

恢复块设计(动态冗余)：动态冗余又称为主动冗余，它是通过故障检测、故障定位及故障恢复等手段达到容错的目的。其主要方式是多重模块待机储备，当系统检测到某工作模块出现错误时，就用一个备用的模块来替代它并重新运行。各备用模块在其待机时，可与主模块一样工作，也可以不工作。前者叫热备份系统(双重系统)，后者叫冷备份系统(双工系统、双份系统)。

防卫式程序设计：是一种不采用任何传统的容错技术就能实现软件容错的方法，对于程序中存

在的错误和不一致性，防卫式程序设计的基本思想是通过在程序中包含错误检查代码和错误恢复代码，使得一旦发生错误，程序就能撤销错误状态，恢复到一个已知的正确状态中去。其实现策略包括错误检测、破坏估计和错误恢复三个方面。

双机容错技术：是一种软硬件结合的容错应用方案。该方案是由两台服务器和一个外接共享磁盘阵列及相应的双机软件组成。

双机容错系统采用“心跳”方法保证主系统与备用系统的联系。所谓心跳，是指主从系统之间相互按照一定的时间间隔发送通信信号，表明各自系统当前的运行状态。一旦心跳信号表明主机系统发生故障，或者备用系统无法收到主系统的心跳信号，则系统的高可用性管理软件认为主系统发生故障，立即将系统资源转移到备用系统上，备用系统替代主系统工作，以保证系统正常运行和网络服务不间断。

工作模式：双机热备模式；双机互备模式；双机双工模式。

合格范文参考

摘要：2021 年 5 月，我所就职的国内某某知名互联网公司组织研发了一套分布式支付平台，该支付平台主要满足公司快速发展和各业务线业务流量日益增加的支付需求，用于支撑各业务线的支付功能，我有幸被定为该平台的架构设计师，主要负责架构设计工作。本文以该支付平台为例，主要论述了软件容错技术和方法在该系统中的具体应用。通过采用以集群化的形式进行应用部署；通过主备形式的数据库部署进行软件容错；通过程序设计方面进行软件的容错与避错。事实证明，以上软件容错技术的应用对于系统的可用性、安全性和可扩展性方面起到了很好的效果，满足了该系统的性能需求，并且该平台从上线到目前一直稳定运行，得到了各业务线负责人和公司高层领导的认可和赞赏。

2021 年 12 月，所就职的国内莫某知名互联网公司组织研发了一套分布式支付平台，由于公司快速发展和各业务线业务流量的日益增加，各业务线迫切需要一个稳定健壮的支付平台，以下简称为平台，用于支撑各业务线的支付功能，公司的业务流量入口分为 PC 端、移动端、微信端和其他渠道等。我作为该平台的架构设计师，主要负责该平台的架构设计工作。

平台采用的核心架构风格为微服务的架构风格，采用 Java 语言为核心开发语言，将平台服务划分为了三类服务，核心服务，平台 Web 服务、平台保障服务。其中核心服务主要分为订单服务、账户服务、网关服务、清结算对账服务、一键支付聚合服务等等，平台 Web 服务主要提供与用户对接的界面等，平台保障服务主要包括例如 JOB 框架，平台报警服务，MQ 服务等等。对这些服务中的核心服务、平台 Web 相关的服务必须要能满足 7*24 小时的稳定运行，对软件的可靠性要求非常高，

所以在该平台中的这些核心服务就必须具备一定的容错能力，在某个服务运行时出错的情况下不能影响到整个集群中的服务，这就要求在软件架构设计中必须考虑到软件容错技术的应用。

提高软件系统可靠性技术主要分为容错技术和避错技术，容错技术的主要方式为冗余，冗余又分为结构冗余、时间冗余、信息冗余，冗余附加。结构冗余又分为静态冗余、动态冗余和混合冗余。软件容错技术主要有 N 版本程序设计，恢复块方法，和防卫式程序设计。结合互联网软件的性质，我主要采用了集群技术、数据库主从方式、和程序设计方面来进行软件的容错与避错处理。下面就从以上三方面详细讨论我所采用的容错技术和方法。

1、通过集群技术来容错

平台中的各服务如果在运行时部署在一台服务器上，那么当服务器发生故障，整个平台将不能再提供任何服务。所以一般非常小规模的应用才会采取这样的部署方式，像互联网应用这样的支付平台来说必须采用多机同时部署的方式，防止单台服务器宕机或者服务进程 Crash 导致整个平台不能提供服务的问题。通过多机同时部署，当一台服务出现问题时，可以很容易替换一台新服务器进行重新部署生效，通过服务客户端的软负载均衡功能，可动态剔除不可用服务机器，动态发现新加入集群的服务机器，使平台在出现故障时可平滑过渡，达到容错的目的。另外平台中各服务当首次获取到的不易变的静态数据会将其存入非本地缓存中，例如采用了 Redis Cluster 技术，可以很好的保证写入缓存中的数据获取的高可靠性，恰当使用缓存不但会提升平台性能，同时还可以起到容错的效果，例如当某个服务所依赖的后端存储发生了短时的故障或者网络抖动，在这个时候大量的并发请求发现存储获取失败直接从缓存中获取将数据返回给调用方，起到了很好的容错效果。

2、通过数据库主从部署方式来容错

对于该平台来说，所依赖的后端数据库存储的稳定性是非常非常重要的，所有的订单，交易、账户等等数据将直接存储到数据库中。如果数据库在运行过程中频繁宕机，那么带来的问题将是不能容忍的，因为会造成订单丢失，交易丢失，账户余额出错等等问题，所以在这里就要求数据库存储要具有非常高的可靠性，同时具有很强的容错性，在这里我主要采用了数据库的部署结构为主从式方式，要求部署在不同服务器上，在不出问题的情况下对于一些时效性要求不是很高的场合从库可以负责承担一部分读流量，当主库发生读写问题时，可快速由其他的从库升级为主库，继续服务，达到容错的效果。在这里我还采用了要求数据库宕机加报警的方式来防止宕机的主从数据库实例过多，导致在并发高的情况下没有可用的从库升级为主库提供服务，通过这样的方式也提高了整个平台的高可靠性。数据库的数据文件存储这里我也要求采用了 RAID 磁盘容错技术来防止单块磁盘损坏导致的数据文件丢失问题。

3、通过程序设计方面进行软件的容错与避错。

根据以往的架构经验，那么系统的不可靠大部分是由于程序内部的设计或者网络请求参数的配置或者连接池参数的配置不当所导致的。所以通过程序设计方面进行软件的容错是非常重要的。在程序设计方面的容错用的最普遍的就是防卫式程序设计，例如平台中的一键支付聚合服务，当在支付的过程中调用账户服务来进行账户金额扣减的时候，势必会调用账户服务传递请求对象来处理，如果说账户服务在被调用的这一刻网络抖动或者丢包的情况下，这个时候一键支付聚合服务必然会收到抛出的错误信息，如果没有通过恰当的容错处理，那么这次一键支付必然会给用户显示支付失败，不太友好，在这里我采用了 TRYCATCH 机制加 3 次重试的容错处理机制，就解决了该次支付因网络抖动导致的支付失败问题。平台采用的是微服务的架构风格，那么在服务之间的通讯过程中涉及到数据的传递，这里我采用了在数据传输协议的头部加 CRC 码来做到对错误数据处理的避错。

通过采用了以上容错技术的方法和措施后，平台从上线运行到目前为止，各服务运行状态良好，通过日志分析来看，支付成功率很高，得到了公司高层领导和各业务线负责人的赞赏和认可。但在我看来还有如下两点不足。1.各服务间调用事物一致性问题的容错处理。针对该问题，目前只能保证事物的最终一致性，因为根据 CAP 理论，要解决该问题确实存在一定的难度，后面我准备研究下 TCC 事物处理方式尤其适合支付平台场景，争取在不损失性能的前提下最大限度解决分布式事物的一致性问题。2.目前所采用的最大努力推送型事物服务依赖 MQ 重复消息的问题。针对该问题我采用加了一张消息处理表的方式来解决，当收到消息的时候，先查询该条消息是否已经处理，如果没有处理直接进行处理并将其进行记录，防止重复处理导致支付数据出错。

软件容错技术对软件的稳定性起着至关重要的作用，尤其是针对互联网性质的软件并发高存在流量峰值等问题，软件容错技术的应用的重要性就不言而喻了。经过这次我所采用的软件容错技术的方法和措施的实施效果后，使我也看到了自己身上的不足之处，我会在今后的架构设计过程中，不断更新自己的知识，不断完善自己的架构设计领域，设计出更好的软件架构，更好的支撑业务平台的运行，提高公司的竞争力，为公司为社会尽一份绵薄之力。

3.9 论软件架构评估

试题 论软件系统架构评估

对于软件系统，尤其是大规模的复杂软件系统来说，软件的系统架构对于确保最终系统的质量具有十分重要的意义，不恰当的系统架构将给项目开发带来高昂的代价和难以避免的灾难。对一个系统架构进行评估，是为了：分析现有架构存在的潜在风险，检验设计中提出的质量需求，在系统被构建之前分析现有系统架构对于系统质量的影响，提出系统架构的改进方案。架构评估是软件开

发过程中的重要环节。

请围绕“论软件系统架构评估”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与架构评估的软件系统，以及在评估过程中所担任的主要工作。

2.分析软件系统架构评估中所普遍关注的质量属性有哪些?详细阐述每种质量属性的具体含义。

3.详细说明你所参与的软件系统架构评估中，采用了哪种评估方法，具体实施过程和效果如何。

理论素材准备

质量属性

1、性能：指系统的响应能力，即要经过多长时间才能对某个事件做出响应，或者在某段时间内系统所能处理的事件的个数。如响应时间、吞吐量。设计策略：优先级队列、增加计算资源、减少计算开销、引入并发机制、采用资源调度等。

2、可靠性：是软件系统在应用或系统错误面前，在意外或错误使用的情况下维持软件系统的功能特性的基本能力。如 MTTF、MTBF。设计策略：心跳、Ping/Echo、冗余、选举。

3、可用性：是系统能够正常运行的时间比例，经常用两次故障之间的时间长度或在出现故障时系统能够恢复正常的速度来表示。如故障间隔时间。设计策略：心跳、Ping/Echo、冗余、选举。

4、安全性：是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。如保密性、完整性、不可抵赖性、可控性。设计策略：入侵检测、用户认证、用户授权、追踪审计。

5、可修改性：指能够快速的以较高的性能价格比对系统进行变更的能力。通常以某些具体的变更为基准，通过考查这些变更的代价衡量。设计策略：接口-实现分类、抽象、信息隐藏。

6、功能性：是系统所能完成所期望的工作的能力。一项任务的完成需要系统中许多或大多数构件的相互协作。

7、可变性：指体系结构经扩充或变更而成为新体系结构的能力。这种新体系结构应该符合预先定义的规则，在某些具体方面不同于原有的体系结构。当要将某个体系结构作为一系列相关产品的基础时，可变性是很重要的。

8、互操作性：作为系统组成部分的软件不是独立存在的，经常与其他系统或自身环境相互作用。为了支持互操作性，软件体系结构必须为外部可视的功能特性和数据结构提供精心设计的软件入口。程序和用其他编程语言编写的软件系统的交互作用就是互操作性的问题，也影响应用的软件体系结构。

敏感点：是指为了实现某种特定的质量属性，一个或多个构件所具有的特性。

权衡点：是影响多个质量属性的特性，是多个质量属性的敏感点。

风险点与非风险点不是以标准专业术语形式出现的，只是一个常规概念，即可能引起风险的因素，可称为风险点。某个做法如果有隐患，有可能导致一些问题，则为风险点；而如果某件事是可行的、可接受的，则为非风险点。

架构权衡分析法 ATAM

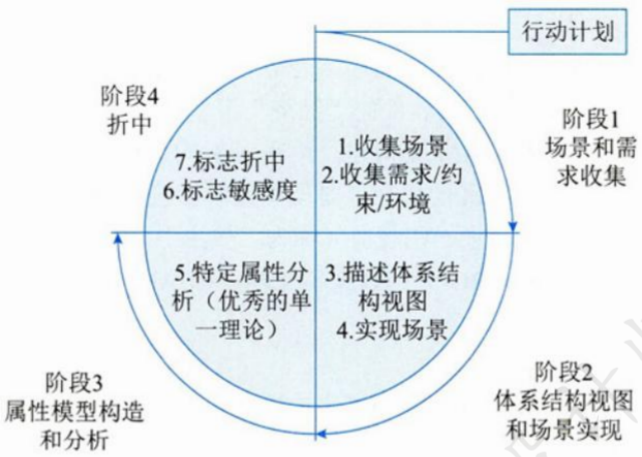


图 8-2　ATAM 分析评估过程

合格范文参考

2023 年 3 月，我公司承担了国家某安全中心漏洞挖掘系统的开发工作，我在该项目中承担系统架构设计师的职务，主要负责系统的架构设计。该项目的主要目的是依托大数据平台从互联网流量中挖掘未知漏洞。本文以漏洞挖掘系统为例，论述了软件系统的架构评估。首先分析了软件架构评估所普遍关注的质量属性并阐述了其性能、可用性、可修改性和安全性的具体含义。整个系统采用了面向服务 SOA 的架构设计方法。在架构设计完成之后，对 SA 评估采用了基于场景的评估方式中的体系结构权衡分析方法 ATAM，并详细描述了其评估过程，项目评估小组经过对项目的风险点、敏感点和权衡点的讨论后生成了质量效应树。目前系统已稳定运行一年多，从而验证了该项目采用 ATAM 架构评估保证了系统的顺利完成。

随着互联网的快速发展，网络上出现的安全问题越来越多，从互联网发展至今，已经爆发了众多的网络攻击事件，如网络蠕虫病毒感染、主机被控制、数据库被非法访问、非法电子银行转账等等。针对这些安全问题，很有必要开发一种 Web 漏洞的发现和利用技术。2022 年 3 月我公司承接了国家某安全中心漏洞挖掘系统的开发工作。该项目通过对互联网中的流量进行特征分析，从中提取出相关的攻击内容，并将这些内容存储到大数据平台，结合大数据分析技术，对攻击者进行跟踪分析，从而捕获出未知漏洞。通过这种漏洞挖掘技术可以极大的解决大数据，大流量背景下 Web 攻击入侵，帮助用户做好“事中”的安全工作，协助安全厂商对互联网攻击进行针对性过滤。

系统在整体架构上采用了面向服务的架构 SOA。前端采用了 PHP 进行开发，后台流量分析工作采用运行性较教高的 C 语言在 Linux 服务器上开发，流量包存储使用了企业磁盘阵列，数据存储采用了 Mysql。通过将系统拆分为多个子模块，各个子模块的构建上用服务进行了封装，它们之间通过消息进行通信。经过对客户需求的分析，我将该系统拆分为了流量捕获模块(负责从互联网中捕获流量)、Pcap 文件存储模块(负责将互联网中的流量存储到大数据平台)、流量分析模块(负责对流量进行分析验证)、数据库模块(负责漏洞数据的存储)和 Web 管理模块(负责下发漏洞规则和查看漏洞信息)。下面先介绍下软件架构评估的质量属性。

架构评估是软件开发过程中的重要环节，在软件架构评估中的质量属性有：性能、可用性、可修改性、安全性、可测试性、可靠性和易用性等。其中前 4 个质量属性是质量效应树的重要组成部分。性能是指系统的响应能力，即经过多长时间对事件做出响应。可用性是指系统能够正常运行的比例，通过用两次故障之间的时间长度或出现故障时系统能够恢复的速度来表示。可修改性是指系统能以较高的性价比对系统做出变更的能力。安全性是指系统能够向合法用户提供服务，同时拒绝非授权用户使用或拒绝服务的能力。

常用的架构评估方法有：基于问卷调查的评估方式、基于场景的评估方式和基于度量的评估方式。基于问卷调查的评估方式是由多个评估专家通过调查问卷的方式回答问卷中的问题，对多个评估结果进行综合，最终得到最终结果。其评价的具有主观性不太适合本项目。基于度量的评估方式虽然评价比较客观，但是需要评估者对系统的架构有精确的了解，也不太适合本项目。而基于场景的评估要求评估者对系统中等了解，评价比较主观，故本项目采用了基于场景的评估方式。基于场景的评估方式又分为架构权衡分析法 ATAM，软件架构分析法 SAAM 和成本效益分析法 CBAM。本项目中根据不同质量属性使用了 ATAM 作为系统架构评估的方法。

在使用 ATAM 进行架构评估时，我们根据项目需要成立了项目评估小组。其主要成员包括：评估小组负责人、项目决策者、架构设计师、用户、开发人员、测试人员、系统部署人员等项目干系人。我在这里的身份是项目的评估小组负责人和首席架构师。架构的评估经历了描述和介绍阶段、调查和分析阶段、测试阶段和报告阶段四个阶段。下面我分别从这四个阶段进行介绍。

在描述和介绍阶段，由于项目评估成员有部分人员对 ATAM 并不熟悉，我首先介绍 ATAM 的方法。它是一种基于场景的软件架构评估方法，对系统的多个质量属性基于场景进行评估。通过该评估确认系统存在的风险，并检查各自的非功能性需求是否满足需求。客户也阐述了系统的目的和商业动机。项目是为了通过捕获互联网流量从而挖掘出有价值的漏洞信息。通过实时获取漏洞可以有效的展开防御，保证网站的安全性。客户关注系统的性能及系统能否获取高质量的漏洞信息。最后作为架构设计师的我描述了系统将要采用的 SOA 架构，并将系统进行了拆分，并讲解了各个子模块

的功能，初步决定系统服务端在 Linux 下使用 C 语言进行开发。

在调查分析阶段，不同的需求方基于各自的考虑都提出了各自的要求。其中客户方提出：系统的要保证其可靠性，特别是针对黑客 IP 进行跟踪的时，系统发生故障必须在 1 分钟内恢复，此优先级最高。经过自动化分析，系统对漏洞的自动识别率必须达到 90%上，此优先级较高。系统可以对规则模块实时进行修改，其修改工作必须在 1 人天完成，以便可以根据最新的规则进行漏洞捕获。系统要确保一定的安全性。安全分析人员提出：系统需要过滤大部分正常的流量，以减轻安全分析人员的分析难度。系统必须提取出有价值的高风险 IP，无效的流量跟踪将会带来产出的低下。开发人员提出为了保证系统的开发效率及系统修改性，可以进行并行开发。经过总结我们获得了系统的质量效应树如下(考试时回简要图)。

针对这些场景我们分析了项目开发过程中的风险点、敏感点和权衡点。经过分析，该项目中存在以下风险点：黑客的 IP 如果不能实时捕获，将会丢失重要漏洞信息；系统中对消息的处理如果超过 12 小时，将会产生大量的消息积压。敏感点有：用户的加密级别、漏洞规则的修改。权衡点有：改变漏洞规则的严格程度会提升漏洞的准确率，同时带来系统性能的下降。改变系统的加密级别对系统的安全性和性能都会产生影响。

在测试阶段：经过评估小组集体讨论，确定了不同场景的优先级如下：系统的可用性最高，性能其次，可修改性及安全性优先级较低。在保证系统可用性方面，在流量捕获部分使用双机热备技术，在两个捕获系统之间设置心跳，当一台捕获系统出问题，另一台捕获设备接管。在流量自动化分析部分，采用了集群部署技术，一台分析设备出问题，不会影响整个分析系统。在保证数据安全性方面，磁盘采用企业磁盘阵列 RAID5 机制。在用户数据安全性方面，采用了非对称加密及信息摘要技术。

最后形成了评估报告，经过对架构的评估，确定了系统的风险点、敏感点、权衡点和非风险点，最后以文档的形式表现。其包括的内容包括：架构分析方法文档、架构的不同场景及各自的优先级、质量效应树、风险点决策、非风险点决策及每次的评估会议记录。

该项目开发工作于 2023 年 9 月完工，系统上线后，我们的安全分析人员和客户使用该系统对互联网流量进行漏洞挖掘，一共产生了 150 种以上的 Web 流量攻击流量特征和 5 个未知 Web 漏洞。在国家某安全中心网研室的其他项目中起到了支撑作用，尤其是某变量覆盖漏洞、某文件写入漏洞，某 SQL 注入漏洞在项目使用过程中取得了一定得效果，得到了好评。为开展互联网安全事件得防御、发现、预警和协调处置等工作提供了数据依据，更好的维护了国家公共互联网安全，保障基础信息网络和重要信息系统的安全运行。

3.10 论信息系统的安全架构设计

试题 论信息系统的安全架构设计

在企业信息化推进的过程中，需要建设许多的信息系统，这些系统能够实现高效率、低成本的运行，为企业提升竞争力。但在设计和实现这些信息系统时，除了针对具体业务需求进行详细的分析，保证满足具体的业务需求之外，还要加强信息系统安全方面的考虑。因为如果一个系统的安全措施没有做好，那么系统功能越强大，系统出安全事故时的危害与损失也就越大。

请围绕“信息系统的安全性与保密性”论题，依次从以下三个方面进行论述：

1.概要叙述你参与分析设计的信息系统及你所担任的主要工作。

2.深入讨论作者参与建设的信息系统中，面临的安全及保密性问题，以及解决该问题采用的技术方案。

3.经过系统运行实践，客观的评价你的技术方案，并指出不足，以及解决方案。

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

参与建设的信息系统中面临的安全及保密性问题。

采用的技术方案。

问题 3 要点：

评价该技术方案。

指出该方案不足并提出解决方案。

理论素材准备

信息安全技术：对称加密，非对称加密，数字信封，信息摘要，数字签名，数字证书。

网络安全技术：防火墙、入侵检测、计算机病毒和木马的防护。

合格范文参考

摘要：2019 年 10 月，我所在公司承接了一款养老管理信息平台，该系统以养老为主线，其中包括养老档案、照护计划、服务审计、状况跟踪、费用管理等方面的 60 多个业务功能模块，我在项目中担任系统架构设计师，主要负责整个系统的架构设计。本文以该养老管理信息平台为例，主要

论述了针对该系统的安全及保密性问题，我们所采用的技术手段及解决方案。在网络硬件层通过设置硬件防火墙，来保证内部网络安全性；在数据层通过设置数据加密及容灾备份机制，以此保证数据抗突发风险的安全；在应用层统一采用 RBAC 的授权机制，来保证整个系统认证环节的安全性。事实证明，以上技术方案的实现对整个系统的安全和保密性方面起到了很好的效果，最终项目顺利上线，运行稳定，受到广大用户的一致好评。

目前我国已经进入到老龄化社会，老龄人口逐年增长，按照老龄办提供的数字，预计到 2020 年中国的老年人口将要达到 2.48 亿，与之增加的养老消费人均三千元左右，从整个养老产业的规模来看，估算在 2025 年要增加到五万亿规模，市场前景巨大。随着互联网的迅猛发展，各行各业都在进行着互联网+的尝试，希望搭上这个发展契机。其中，养老领域更迫切需要解决养老专业化程度低，信息化不足，健康照护水平滞后等一系列亟待解决的问题。

2019 年 10 月，我所在公司承担了全国老龄办及全国几十家养老和医疗机构合作进行的养老管理信息平台的开发工作，我在该项目中担任系统架构设计师，主要负责应用系统的软件架构设计。由于我们公司在医疗行业领域有着丰富的成功经验，同时，近些年在养老领域也成功实施过很多成熟的案例，所以，一期投资方出资 3000 万，委托我们进行这款综合性养老管理平台的开发工作。该系统以养老为主线，其中包括养老档案，照护计划，服务审计，状况跟踪，费用管理，决策支持等方面的 60 多个业务功能模块组成，系统功能相当完备。

经过前期对全国几十家养老机构和相关合作的医疗单位的调研分析，结合原有的经验，对整个系统业务进行详细规划和相关设备的初步选型，即我们内部所说的“两版三端”的方案，机构养老和社区养老两个子系统，同时能够在 PC 端，PAD 端，手机端三端进行呈现和交互。整个养老管理信息平台基于 B/A/S 模式的分层架构风格设计。同时，结合当前先进的基于租户理念搭建的一个养老云平台(SAAS)。当前信息安全问题严重，而养老云平台是一个养老大型综合管理平台，确保系统的安全性和保密性显得尤为重要，让使用我们平台的每一家机构、社区、老人及相关亲人都能够安心和放心。这样，只有最终得道了用户的信赖，系统的前景才会发展的更好。所以，我在系统的安全性和保密性方面做了充分的设计。

1、网络硬件层安全方案

首先，因为网络和硬件是整个系统运行的基础，也是很多外部攻击的主要途径，所以，在这块需要进行合理规划。我们对网络拓扑结构划分为外部网络，内部网络和 DMZ 三部分。在外部网络和内部网络之间设置了硬件防火墙，主要是防止外部的恶意攻击。在防火墙后，为了加强对病毒入侵的防范，又设置了硬件的防毒墙，实时保证最新的病毒特征库保证有效的病毒工具。为了防止外部客户通过 DNS 服务直接访问到我们服务器的目标地址，在内部网络中又增加了反向代理服务器，

对外只暴露代理服务器的虚拟 IP，更好的保护了应用服务器被攻击的可能性。应用服务器和数据库服务器做了物理隔离，内部人员也无法通过 IP 直接访问，只能通过跳板机由服务器管理员进行操作，阻断了外界通过内部客户端代理的访问对服务器造成攻击。对于一些核心的 FTP 服务器，DNS 服务器，Web 服务器都规划到 DMZ 中。

2、数据层安全方案

其次，数据是整个系统平台的核心，其中涉及大量个人隐私数据和重要信息资产，安全不容忽视。主要从数据存储，数据访问和数据容灾几方面进行了规划。由于系统主要存放的养老档案数据，其中涉及很多个人隐私部分数据，为了防止泄露，我们在存储时进行了加密处理。虽然在处理中损失些性能，但是，提高了数据的安全性和保密性。我们采购商业数据库 Oracle 作为后台存储，由于其极佳的商业口碑，对数据存储的安全提供强大保证。为了防止数据信息的泄露，加强了访问权限的限制，在数据库的访问权限上，都根据不同角色进行了详细划分，包括对数据库、表、索、记录等的增、删、改、查进行了限制，严格保证非法用户对数据库恶意破坏。同时，在数据库本身，也制定了基于全量、增量、差量的按周备份计划，保证每时每刻的数据都可以及时恢复。在物理存储上也做了本地多机房的容灾备份，充分保证数据抗突发风险的安全。

3、应用层安全方案

最后，由于我们系统架构采用的是分层架构风格，同时，又是采用“三端”的形式，即手机端，PAD 端，还有 PC 端，所以，在用户认证，接口传输等方面做了充分设计。由于系统涉及到养老机构工作人员，老人及家属等，角色众多，统一采用 RBAC 的授权机制，既增强系统安全性满足业务需求，同时，减轻由于权限信息维护负担。在登录界面设计上，不仅仅需要提供用户名+口令，同时设置了校验码，而且同时还增加了动态口令，通过短信接收验证码，以此来防止暴露非法破解登录，保证整个系统认证环节的安全性。密码提交和存储过程中，一律通过 MD5+SALT 加密，即使密码泄露也不会被解密。接口设计是整个系统通信的安全保障，为了防止外部人员恶意调用和窃取信息，我们采用了令牌机制保障每次通讯的安全性，即在登录后，通过加密传输的用户信息和时间戳获取到 Token 令牌，然后在后续每次通讯传输过程服务端对 Token 进行校验，充分保障了接口调用的安全。

整个项目历时 10 个月开发完成，到目前运行稳定。通过用户一段时间的使用反馈，不论是 PC，还是 PAD 和手机端，在任何环境下使用系统，基本没有信息泄露的情况发生。但是，一些用户反馈，手机登录需要时间较长，查询和保存老人档案信息有时候需要等待，经过分析，登录时间较长，由于登录需要对客户端对用户和密码信息进行客户端加密处理，加密算法耗费了移动端 CPU 资源，同时在服务端也对数据进行加密处理返回客户端 Token 也消耗了不少时间。老人档案也是由于敏感信

息加密的问题，耗费了 CPU 的处理时间，我们后面通过选择效率更高的加密算法改善了这个问题。

实践证明，项目能够顺利上线，并运行稳定，使用良好，与系统在安全性和保密性方面设计密不可分。系统安全是一个永久的话题，我们对系统安全性的完善是一个持续的过程，我们接下来，还会继续不断完善系统安全方面的设计缺陷和不足，使整个养老平台更加安全好用。

3.11 基于构件的软件开发

试题 基于构件的软件开发

软件系统的复杂性不断增长、软件人员的频繁流动和软件行业的激烈竞争迫使软件企业提高软件质量、积累和固化知识财富，并尽可能地缩短软件产品的开发周期。

集软件复用、分布式对象计算、企业级应用开发等技术为一体的“基于构件的软件开发”应运而生，这种技术以软件架构为组装蓝图，以可复用软件构件为组装模块，支持组装式软件的复用，大大提高了软件生产效率和软件质量。

请围绕“基于构件的软件开发”论题，依次从以下三个方面进行论述。

1.简述你所参与开发的运用了构件技术的项目，以及你所担任的工作；

2.论述你在项目中如何运用构件技术来进行软件开发；

3.分析并讨论各种构件技术的优点、缺点，并展望构件技术的发展趋势。

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

如何运用构件技术来进行软件开发。

具体实施过程。

问题 3 要点：

各种构件技术的优点、缺点。

展望构件技术的发展趋势。

理论素材准备

构件是一个独立可交付的功能单元，外界通过接口访问其提供的服务。

构件由一组通常需要同时部署的原子构件组成。一个原子构件是一个模块和一组资源。原子构

件是部署、版本控制和替换的基本单位。原子构件通常成组地部署，但是它也能够被单独部署。

构件和原子构件之间的区别在于，大多数原子构件永远都不会被单独部署，尽管它们可以被单独部署。相反，大多数原子构件都属于一个构件家族，一次部署往往涉及整个家族。

一个模块是不带单独资源的原子构件(在这个严格定义下，Java 包不是模块——在 Java 中部署的原子单元是类文件。一个单独的包被编译成多个单独的类文件——每个公共类都有一个)。

模块是一组类和可能的非面向对象的结构体，比如过程或者函数。

构件和对象

构件的特性是：(1)独立部署单元；(2)作为第三方的组装单元；(3)没有(外部的)可见状态。一个构件可以包含多个类元素，但是一个类元素只能属于一个构件。将一个类拆分进行部署通常没什么意义。

对象的特性是：(1)一个实例单元，具有唯一的标志。(2)可能具有状态，此状态外部可见。(3)封装了自己的状态和行为。

构件接口

接口标准化是对接口中消息的格式、模式和协议的标准化。它不是要将接口格式化为参数化操作的集合，而是关注输入输出的消息的标准化，它强调当机器在网络中互连时，标准的消息模式、格式、协议的重要性。

面向构件的编程(COP)

关注于如何支持建立面向构件的解决方案。“面向构件的编程需要下列基本的支持：

——多态性(可替代性);

——模块封装性(高层次信息的隐藏);

——后期的绑定和装载(部署独立性);

——安全性(类型和模块安全性)。

构件技术就是利用某种编程手段，将一些人们所关心的，但又不便于让最终用户去直接操作的细节进行了封装，同时对各种业务逻辑规则进行了实现，用于处理用户的内部操作细节。

目前，国际上常用的构件标准主要有三大流派，分别是 COM/DCOM/COM+，CORBA 和 EJB。

EJB(Enterprise Java Bean)规范由 Sun 公司制定，有三种类型的 EJB，分别是会话 Bean(Session Bean)、实体 Bean(Entity Bean)和消息驱动 Bean(Message-driven Bean)。

EJB 实现应用中关键的业务逻辑，创建基于构件的企业级应用程序。EJB 在应用服务器的 EJB 容器内运行，由容器提供所有基本的中间层服务，如事务管理、安全、远程客户连接、生命周期管理和数据库连接缓冲等。

COM、DCOM、COM+：COM 是微软公司的。DCOM 是 COM 的进一步扩展，具有位置独立性和语言无关性。COM+并不是 COM 的新版本，是 COM 的新发展或是更高层次的应用。

COBRA 标准主要分为三个层次：对象请求代理、公共对象服务和公共设施。

最底层是对象请求代理 ORB，规定了分布对象的定义(接口)和语言映射，实现对象间的通讯和互操作，是分布对象系统中的“软总线”;

在 ORB 之上定义了很多公共服务，可以提供诸如并发服务、名字服务、事务(交易)服务、安全服务等各种各样的服务；

最上层的公共设施则定义了组件框架，提供可直接为业务对象使用的服务，规定业务对象有效协作所需的协定规则。

合格范文参考

摘要：本文以我主持开发的某公司生产经营管理系统为例，探讨了基于构件的软件开发问题。该系统是一个集原料采购、生产管理、物流管控等七大功能于一体的综合信息系统，在该系统的开发过程中，我担任系统架构师角色，主要负责需求分析、系统建模和方案设计三个方面的工作。本文首先简要分析了 CORBA、EJB、COM/DCOM 三种构件技术的特点，然后着重论述了采用构件技术进行软件开发的过程。在构件的获取阶段，我们采用了三种构件获取方式来解决用户提出的 3 类不同的需求；在构件的开发阶段，我们统一采用一个查询构件进行了封装，以实现将同一功能的不同表现封装到一个独立的构件中；在构件组装阶段，采用了 3 种构件组装方式完成了构件的组装。最终项目顺利上线并运行稳定，获得用户一致好评。

2023 年 5 月，我所在公司承接了某大型粮食加工企业某公司生产经营综合管理系统的开发工作。该系统是某公司在国家粮食案例政策的指导下，结合自身经营管理需求提出建设的。其目的是对内加强管理，对外提升服务，以实现提升品牌形象、保护消费者利益的战略目标。系统整体上分为两个部分，一是经营管理 Web 平台，二是手机 APP 应用。系统采用了基于服务的层次架构，共分为三层。其中用户界面层使用 Extjs、Senchatouch 和 Phonegap 框架实现，业务服务层使用.Net 平台实现，数据层使用 IBMDB2V9.5。该系统于 2023 年 5 月开始建设，2024 年 6 月上线运行，历时一年。在系统的开发过程中，我担任系统架构师角色，负责需求的获取和分析、系统建模和总体方案设计工作。在系统的实现过程中，我和我的团队使用了基于构件的软件开发技术，收到了很好的效果。

基于构件的软件开发是在面向对象技术的基础上发展起来的，它是软件危机问题日益突出形势下的产物，它有效地解决了软件系统复杂度、成本、质量、效率等难以控制的问题，受到业界的广泛推崇。当前主流的软件构件技术有三大流派，分别是 OMG 的 CORBA、SUN 的 EJB 和微软的 COM

技术。上述三种技术中，CORBA 实现的是最为漂亮的，它分为对象请示代理、公共对象服务和公共设施三个层次，其特点是大而全，互操作性和开放性特别好，其缺点是庞大且复杂，相关技术和标准更新缓慢。相比之下，EJB 的发展要更加迅速，它是在 Java 语言基础上建立的服务器端组件模型，具有优秀的跨平台性。EJB 框架提供了远程访问、安全、持久化和生命周期等多种支持分布式计算的服务，目前 JAVA 和 CORBA 有融合的趋势。最后是 COM 技术，它是微软公司的独家产品，基于 Windows 平台，功能强大、效率很高，且有一系列相应的开发工具支持，其最大的弱点是跨平台性较差，很多人认为 CORBA 将比 COM 技术走地更远。基于构件的软件开发技术能有效地简化设计、提高效率、保证质量，某公司生产经营管理系统的开发项目时间紧、任务重，我们使用基于构件的软件开发技术解决了软件开发中的各种问题，取得了很好的效果。

具体在实践过程中，我们开展了模块划分、构件标识、构件获取、构件组装与测试、构件管理等活动，这里重点谈一下“构件获取”、“构件开发”和“构件组装”三个方面的问题。

一、构件获取

在本系统的开发过程中，用户提出的需求从实现方式上可分为三类。第一类需要修改当前的系统来实现，比如用户提出，需要将某公司当前在用的两套生产控制系统的支行状态集成到目标系统中；第二类是直接使用我公司现成的构件来实现。比如用户登录、角色权限管理、日志记录、数据库访问等基本功能；第三类需要集成第三方的服务来实现，比如用户要求 APP 软件需要具有消息推送、GPS 定位功能，可通过集成第三方运营商的产品来实现，无需另行开发。针对上述三种类型的需求，我们采用了三种构件获取方式。首先是通过改造现有系统获得构件，我们与生产控制系统的开发商联系，请他们使用 COM 技术对软件的工作状态进行封装，并开放接口。其次是使用构件库中现成的构件，我公司在长期的软件项目建设过程中，积累了大量可重用的软件构件，如文件序列化、数据库连接等模块，这些构件可直接在新项目中使用。最后是集成第三方的构件，针对用户提出的消息推送和 GPS 定位功能，我们经过对比和测试，选择了个百度的产品进行集成。此外我们还根据需求，重新开发了一些功能构件，以支撑用户的需求。

二、构件开发

构件的优势体现在其粗粒度的重用性，因此在构件的设计过程中，应尽可能将同一功能的不同表现封装到一个独立的构件中，以保持其高内聚、低耦合的特性，本系统的构件设计就很好地遵循了这一原则。以数据查询构件为例，整个系统中用户需要查询的数据多种多样，对分页显示的要求不尽相同，数据返回格式也不完全一致，可能是 XML，也可能是 JSON，针对所有这些查询需求，我们统一用一个查询构件进行了封装，开发人员只需要构造好 SQL 语句，再配合一些特定的参数，就能得到自己想要的结果，这样最大限度地保证了构件的可重用性和重用粒度。为了实现这一目标，

有时还需要用到一些经典的设计模式。比如我们使用的数据库连接构件，就用到了抽象工作方法，构件可根据配置文件的内容，在运行过程中建立对不同数据库、不同方式的连接，很好地体现了构件的优势。

三、构件的组装

不同的构件类型，需要采用不同的组装方式。在本系统的开发过程中，我们用到了以下三种方式。首先是 DCOM 构件的组装，采用了远程调用方式，该方式主要用于对生产控制系统构件的组装。其实现比较复杂，需要在控制系统上部署 DCOM 服务，还得在 WEB 平台的 IIS 上配置相应的权限。其次是构件库中构件的组装，该方式比较简单，直接引用构件库中的产品文件，可能是 DLL 格式，或者是 C#的源码文件。最后是 SOA 服务调用组装，主要针对第三方的软件服务，其实现也很简单，使用基于 Http 的 Webservice 访问即可。需要注意的是在这种组装方式中，由于数据需要经过第三方的软件平台进行传输，需要采用一定的数据加密措施，以提高系统的安全性。

由于使用了基于构件的软件开发方式，某公司生产经营管理系统的开发工作进行地十分顺利，系统按期上线，并得到了用户的肯定。该项目的成功让我认识到好的软件设计思想和技术在软件开发过程中的作用和价值，坚定了我对基于构件的软件开发技术的信心。从软件行业的发展来看，从汇编语言、基于过程的软件开发、面向对象的软件开发，直到现在基于构件、面向服务的软件开发，我们可以认识到软件元素在两个维度上的进化趋势，即内部功能越来越强大、全面，对外的接口却越来越简单、标准，终有一日各领域的软件必将能在一个统一的标准下进行无缝的组装，届时面向协作的软件开发、基于职能的软件开发等新技术都可能会出现，上层应用功能的实现也将变地异常简单，计算机软件可能会变地无所不在，数字化生活、智能地球等等现在还比较超前的概念，将在那里得以实现。我想这一天值得每一位软件从业人员为之努力、期待。

3.12 论企业集成平台的技术与应用

试题 论企业集成平台的技术与应用

企业集成平台是一个支持复杂信息环境下信息系统开发、集成和协同运行的软件支撑环境。它基于各种企业经营业务的信息特征，在异构分布环境(操作系统、网络、数据库)下为应用提供一致的信息访问和交互手段，对其上运行的应用进行管理，为应用提供服务，并支持企业信息环境下各特定领域的应用系统的集成。企业集成平台的核心是企业集成架构，包括信息、过程、应用集成的架构。

请以“企业集成平台的技术与应用”为题，依次从以下三个方面进行论述：

1.概要叙述你参与管理和开发的企业集成平台相关的软件项目以及你在其中所担任的主要工作。

2.简要说明企业集成平台的基本功能及企业集成的关键技术，并结合项目实际情况，阐述该项目所选择的关键技术及其原因。

3.结合你具体参与管理和开发的实际项目，举例说明所采用的企业集成架构设计技术的具体实施方式及过程，并详细分析其实现效果。

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

企业集成平台的基本功能及企业集成的关键技术。

结合项目实际情况，阐述该项目所选择的关键技术及其原因。

问题 3 要点：

采用的企业集成架构设计技术的具体实施方式及过程。

详细分析其实现效果。

理论素材准备

企业集成分类

（1）表示集成：即界面集成，是最原始的集成，黑盒集成。将多个信息系统的界面集成在一起，统一入口，为用户提供一个看上去统一，但是由多个系统组成的应用系统的集成，例如桌面。

（2）数据集成：白盒集成，把不同来源、格式、特点性质的数据在逻辑上或者物理上有机的集中，从而为企业提供全面的数据共享。如数据仓库。

（3）控制集成(功能集成、应用集成):黑盒集成，业务逻辑层次的集成，可以借助于远程过程调用或远程方法调用、面向消息的中间件等技术，将多个应用系统功能进行绑定，使之像一个实时运行的系统一样接受信息输入和产生数据输出，实现多个系统功能的叠加。如钉钉。

（4）业务流程集成：即过程集成，最彻底的、综合的集成，这种集成超越了数据和系统，由一系列基于标准的、统一数据格式的工作流组成。当进行业务流程集成时，企业必须对各种业务信息的交换进行定义、授权和管理，以便于改进操作、减少成本、提高响应速度。它包括应用集成、B2B 集成、自动化业务流程管理、人工流程管理、企业门户，以及对所有应用系统和流程的管理和监控等。如电子购物网站-第三方支付平台-银行-物流等流程集成。

应用集成数据交换方式

共享数据库：在应用集成时，让多个应用系统通过直接共享数据库的方式，来进行数据交换，实时性强，可以频繁交互，属于同步方式；但是安全性、并发控制、死锁等问题突出。

消息传递：消息是软件对象之间进行交互和通信时所使用的一种数据结构，可以独立于软件平台而存在，适用于数据量小、但要求频繁、立即、可靠、异步地数据交换场合。

文件传输：是指在进行数据交换时，直接将数据文件传送到相应位置，让目标系统直接读取数据，可以一次性传送大量信息，但不适合频繁进行数据传送。适用于数据量大、交换频度小、即时性要求低的情况。

企业集成平台

集成平台是支持企业集成的支撑环境，包括硬件、软件、软件工具和系统，通过集成各种企业应用软件形成企业集成系统。由于硬件环境和应用软件的多样性，企业信息系统的功能和环境都非常复杂，因此，为了能够较好地满足企业的应用需求，作为企业集成系统支持环境的集成平台，其基本功能主要有：

（1）通信服务它提供分布环境下透明的同步/异步通信服务功能，使用户和应用程序无需关心具体的操作系统和应用程序所处的网络物理位置，而以透明的函数调用或对象服务方式完成它们所需的通信服务要求。

（2）信息集成服务它为应用提供透明的信息访问服务，通过实现异种数据库系统之间数据的交换、互操作、分布数据据管理和共享信息模型定义(或共享信息数据库的建立)，使集成平台上运行的应用、服务或用户端能够以一致的语义和接口实现对数据(数据库、数据文件、应用交互信息)的访问与控制。

（3）应用集成服务它通过高层应用编程接口来实现对相应应用程序的访问，这些高层应用编程接口包含在不同的适配器或代理中，它们被用来连接不同的应用程序。这些接口以函数或对象服务的方式向平台的组件模型提供信息，使用户在无需对原有系统进行修改(不会影响原有系统的功能)的情况下，只要在原有系统的基础上加上相应的访问接口就可以将现有的、用不同的技术实现的系统互联起来、通过为应用提供数据交换和访问操作，使各种不同的系统能够相互协作。

（4）二次开发工具二次开发工具是集成平台提供的一组帮助用户开发特定应用程序(如实现数据转换的适配器或应用封装服务等)的支持工具，其目的是简化用户在企业集成平台实施过程中(特定应用程序接口)的开发工作。

（5）平台运行管理工具它是企业集成平台的运行管理和控制模块，负责企业集成平台系统的静态和动态配置、集成平台应用运行管理和维护、事件管理和出错管理等。通过命名服务、目录服务、平台的动态静态配置，以及其中的关键数据的定期备份等功能来维护整个服务平台的系统配置及稳

定运行。

合格范文参考

摘要：2023 年 2 月，本人所在的某家商业银行启动了零售 CRM 项目建设，该项目主要实现客户管理、客户分析、营销管理、绩效管理、数据 ETL 处理、多渠道数据服务等功能，在此项目中，我担任架构师，负责项目总体架构设计工作。本文以该零售 CRM 项目建设为例，主要论述了企业集成架构设计技术在该系统中的具体应用。操作型 CRM 和分析型 CRM 两个子系统通过零售 CRM 门户实现界面集成；数据服务系统通过银行已有 ESB 与各请求系统实现应用集成；利用数据总线和数据仓库实现数据集成，建立仓内零售 CRM 数据集市，为客户分析、绩效管理、数据服务提供数据支持。通过以上集成架构技术将项目中多个子系统进行整合，形成有机整体，最终项目顺利上线，至今系统运行稳定，受到用户一致好评。

本人所在的某家商业银行分支机构遍布全国省会和重点城市，零售客户数量近 6 千万，客户经理数量约 5 千。客户经营维系情况参差不齐，客户经理经营维系客户的策略、方法、工具缺乏统一，抬高了经营成本，降低了客户体验。营销活动的有效性和目标性不强，往往营销成本较大但效果不明显，针对这些问题，同时为了适应竞争激烈的市场环境，该银行制定了零售 CRM 业务战略，并于 2023 年 2 月启动了零售 CRM 项目建设，主要实现客户管理、客户分析、营销管理、绩效管理、数据 ETL 处理、多渠道数据服务等功能，该项目旨在建立全行统一的零售客户经营管理体系和零售数据分析环境，充分发挥长期积累数据的业务价值，科学、有效的指导客户获取、营销、服务、挽留等全生命周期活动，提高客户经营和服务水平，增强客户体验，提升客户价值，从而提高零售业务的利润回报。

在此项目中，我担任架构师，负责项目总体架构设计。项目涉及范围广泛，我采用“分而治之”策略，根据应用类型、用户角色、数据处理特征的不同，将项目分为多个(子)系统，并研究企业集成的关键技术，根据项目实际情况，选择合适的集成技术并进行架构设计。

企业集成是指使用应用服务器、中间件等平台和技术，连接企业内的各应用系统，实现异构系统之间的交互和协作，以及数据交换和共享。企业集成模式与技术主要有以下四种：1、界面集成，把各个应用系统的界面集中在一个界面之中，常用技术为企业门户。2、过程集成，使各个应用系统连接起来支持完整的业务流程，常用技术为工作流、企业门户。3、应用集成，为两个以上应用系统中的数据和程序提供接近实时的集成，常用技术为远程过程调用、消息中间件、服务总线(ESB)、Web 服务等。4、数据集成，解决的是信息系统之间数据同步(包括主数据系统向副本系统同步，源系统向数据仓库同步)和时效性(包括实时、批量)问题，常用技术为适配器、消息中间件、数据总线

数据仓库。下面我将以通过企业门户、服务总线、数据总线、数据仓库等技术或平台，来实现界面、应用、数据集成来具体论述其实现过程。

1、操作型 CRM 和分析型 CRM 两个子系统通过零售 CRM 门户实现界面集成。

客户管理、营销管理、客户分析、绩效管理四部分功能，属于面向银行内部用户的管理分析类需求，适宜采用 BS 架构搭建零售 CRM 系统。为了降低耦合，将零售 CRM 系统划分为操作型 CRM(实现客户管理和营销管理功能)和分析型 CRM(实现客户分析、绩效管理功能)两个子系统。但对于用户使用便捷性却不能降低，需要将两个子系统的界面集中在一起，这要使用界面集成技术，于是开发零售 CRM 门户进行界面集成。零售 CRM 门户作为零售用户的统一工作台，为两个子系统提供统一登录界面和访问入口、统一用户认证和授权、统一界面风格、统一功能菜单和待办事项，使用代理技术实现门户系统向两个子系统页面的路由和跳转，提高了用户体验，使得子系统划分对于用户透明。门户系统采用 J2EE 开源框架(Spring、MyBatis)进行开发，技术架构分为页面展现、接入控制、业务逻辑、数据访问四层。

2、数据服务系统通过银行已有 ESB 与各请求系统实现应用集成。

多渠道数据服务功能，既面向客户经理通过零售 CRM 系统 Web 界面查询客户交易数据，也面向客户通过网上银行和手机银行系统查询交易数据。经分析，查询功能可以定义为标准的服务，支持复用且面向多个请求系统提供接口访问，各系统技术异构，适宜采用 SOA 架构风格搭建数据服务系统，并使用应用集成技术，利用银行已有服务总线(ESB)向多个系统提供查询服务。ESB 包括服务管理、协议转换、格式转换、服务路由、消息处理等功能。将数据服务系统的查询服务在 ESB 中注册并发布，各请求系统向 ESB 发起服务请求，ESB 接入请求，通过适配器技术进行通讯协议和报文格式的转换，经服务路由，向数据服务系统提交查询请求。

3、利用数据总线和数据仓库实现数据集成。

数据 ETL 处理，指将银行前台存款、贷款、信用卡、理财等上百个、技术异构、数据标准不统一的交易系统的数据进行采集、转换、清洗、存储、整合、加工，为客户分析、绩效管理、数据服务提供数据支持，这正是数据总线和数据仓库要解决的问题，属于数据集成范畴。数据总线设计过程为：通过数据采集器(IBMII 产品)，每日日终自动从源系统采集数据并将编码转换为文本格式，通过数据处理器(IBM Data Stage 产品)，进行数据质量检查(要定义检查规则)、增量数据生成、数据标准转换等(要定义数据标准)处理，通过数据传输器，将数据文件传输给目标系统(包括数据仓库)。数据仓库设计过程为：接收数据总线传输来的数据，加载至数据仓库系统(要选择具有海量数据处理能力且能水平扩展的产品)，对数据按照主题模型(比如客户、产品、账户、交易、渠道等)进行归类、整合，并存放长期历史数据，之上建立零售 CRM 数据集市，根据本次需求的数据加工规则，对数

据进行关联、聚合等加工处理，生成数据文件，传输并加载至分析型 CRM 子系统、数据服务系统，为客户分析、绩效管理、交易查询提供数据支持。

项目经过一年时间的开发测试，于 2024 年 2 月上线运行，至今系统运行稳定，使用效果良好，有力地支持了零售数据分析和客户经营管理。项目总结会上，领导对于系统划分方法及采用的界面、应用、数据集成方式表示高度认同。

近来，业务提出呼叫中心客服坐席要查询客户历史交易明细，由于采用了 ESB 应用集成技术，所以可以方便接入呼叫中心系统。面对客户分析和绩效管理的数据维度和指标经常发生变化且要重算，由于数据仓库积累了长期历史数据，所以方便应对数据变更需求。但面对客户提出通过网上银行和手机银行系统查询当日发生的交易明细需求，由于目前数据总线只支持数据 T+1 时效，所以要在数据总线中增加实现数据准实时采集机制，将交易数据准实时同步至数据服务系统，以提高数据服务时效性。

第 4 章　历年论文真题及写作要点

4.1 2024 上半年

试题一　论大数据 Lambda 架构

大数据处理架构是专门用于处理和分析巨量复杂数据集的软件架构。它通常包括数据收集、存储、处理、分析和可视化等多个层面，旨在从海量、多样化的数据中提取有价值的信息。Lambda 架构是大数据平台里最成熟、最稳定的架构，它是一种将批处理和流处理结合起来的大数据处理系统架构，其核心思想是将批处理作业和实时流处理作业分离，各自独立运行，资源互相隔离，解决传统批处理架构的延迟问题和流处理架构的准确性问题。

请围绕“大数据处理架构及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及你在其中所承担的主要工作。

2.Lambda 体系结构将数据流分为三个层次：批处理层(Batch Layer)、加速层(Speed Layer)和服务层(Serving Layer)，请简要分析这三个层次的特性和用途。

3.具体阐述你参与管理和开发的项目是如何基于 Lambda 架构实现大数据处理的。

试题一 写作要点

撰写关于 Lambda 架构的软考论文时，一个清晰且结构化的大纲是成功的关键。以下是一个简单的论文大纲示例，旨在覆盖 Lambda 架构的核心概念、设计原则、优缺点、实际应用案例以及对比其他架构(如 Kappa 架构)的分析：

大纲

●简要介绍 Lambda 架构的基本概念及其在大数据处理领域的地位。

●概述论文的主要研究内容、目的及预期贡献。

●背景介绍：阐述大数据处理的挑战，特别是在实时性和历史数据一致性方面。

●研究意义：解释研究 Lambda 架构的重要性，尤其是在现代数据驱动型企业中的应用。

●论文结构概述。

结合项目可以写的一些内容

论述 1：Lambda 架构概述

●定义：明确 Lambda 架构的定义及提出背景。

●架构层次：详细介绍批处理层、加速层和服务层的功能与作用。

●设计原则：概括 Lambda 架构遵循的核心设计理念。

论述 2：Lambda 架构的核心组件与工作原理

●批处理层(Batch Layer)：描述其在数据摄入、处理和建立不可变数据视图的作用。

●加速层(Speed Layer)：分析实时处理和补充最近数据的功能。

●服务层(Serving Layer)：说明如何合并实时与批处理数据，提供统一查询接口。

●数据流与一致性：探讨数据在不同层间流动的机制及确保查询结果一致性的策略。

论述 3：Lambda 架构的优势与局限性

●优势：即时查询能力、容错与可扩展性、兼容历史数据处理。

●局限性：数据重复处理、维护复杂度高、系统设计复杂。

论述 4：Lambda 架构的实际应用案例分析

●选取一至两个行业或具体项目，分析 Lambda 架构的应用场景、实施效果与面临的挑战。

●例对比：在不同规模和需求下的适用性讨论。

论述 5：Lambda 架构与其他架构的对比(可选，具体看论文题目有没有)

●Kappa 架构介绍：概述 Kappa 架构的设计理念和工作流程。

●对比分析：从数据处理模式、实时性、维护成本等方面对比 Lambda 与 Kappa。

●选择考量：根据应用场景指导何时选择 Lambda 或 Kappa。

论文结尾：

●总结 Lambda 架构在项目实践过程中的使用效果，以及对于项目的核心价值和存在的问题。

●强调其在大数据处理领域的持续影响力，并指出研究的现实意义。

试题二 论模型驱动架构设计方法及其应用

模型驱动架构设计是一种用于应用系统开发的软件设计方法，以模型构造、模型转换和精化为核心，提供了一套软件设计的指导规范。在模型驱动架构环境下，通过创建出机器可读和高度抽象的模型实现对不同问题域的描述，这些模型独立于实现技术，以标准化的方式储存，利用模型转换策略来驱动包括分析、设计和实现等在内的整个软件开发过程。

请围绕“模型驱动架构设计方法及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与分析、设计的软件项目以及你在其中所承担的主要工作。

2.请简要描述采用模型驱动架构思想进行软件开发的全过程及其特点。

3.具体阐述你参与的软件项目是如何基于模型驱动架构完成分析、设计和开发的。

试题二 写作要点

领域驱动设计(Domain-Driven Design，DDD)与模型驱动软件开发方法(Model-Driven Software Development，MDSD)是两种注重软件开发质量和效率的方法论。下面分别概述两者的核心概念，并探讨它们如何相互作用以提升软件项目的成功概率。

（1）**领域驱动设计(DDD)**

核心要素：

1.核心领域：识别并聚焦于业务中最关键、最具价值的部分。

2.界限上下文：划分不同的领域边界，确保领域内的模型一致性。

3.领域模型：通过实体、值对象、聚合根等元素精确表达业务逻辑。

4.通用语言：建立业务和技术团队间的共同语言，减少误解。

5.分层架构：常见的有战略设计、战术设计分层，确保领域逻辑的纯净性。

（2）**模型驱动软件开发方法(MDSD)**

基本概念：

1.抽象模型：从不同的视角(如业务、技术)建立软件的抽象表示。

2.模型转换：使用转换引擎或代码生成工具将高级模型转换为可执行代码或配置。

3.元模型与元数据：定义模型的结构和规则，支持模型的标准化和复用。

4.工具支持：依赖于建模工具、代码生成器等自动化工具链来提高开发效率。

5.迭代与反馈：模型不是静态的，需在开发过程中不断迭代优化，以反映真实需求。

DDD 与 MDSD 的协同

结合点：

●领域模型作为起点：DDD 的领域模型可作为 MDSD 中高层次抽象模型的基础，确保模型直接反映业务本质。

●模型驱动的领域实现：利用 MDSD 工具将 DDD 中定义的领域模型转换为具体的实现代码，提高开发效率，减少错误。

●复杂性管理：DDD 帮助识别和管理软件的复杂性，而 MDSD 通过模型的自动化转换减少实现层面的复杂性。

●迭代与进化：DDD 强调通过迭代深化对领域的理解，而 MDSD 支持模型的快速迭代和演化，二者结合促进了软件的持续优化。

●标准化与复用：MDSD 的元模型和元数据管理有助于 DDD 中核心领域模型的标准化表述和跨项目复用。

结论

结合领域驱动设计与模型驱动软件开发方法，可以构建更加贴合业务需求、易于维护和扩展的软件系统。DDD 确保软件设计深度契合业务领域，而 MDSD 则通过自动化工具链提升开发效率和质量，两者相辅相成，共同促进软件工程实践的现代化和高效性。在实践中，需要根据项目特点灵活运用这两种方法，平衡业务理解的深度与技术实现的效率。

试题三 论单元测试及运用

请围绕“论单元测试及运用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及你在其中所承担的主要工作。

2.结合你参与管理和开发的软件项目，简要叙述单元测试中静态测试和动态测试方法的基本内容。

3.结给你参与管理和研发的软件项目，阐述在测试过程中，如何确定白盒测试的覆盖标准，及如何组织实施回归测试。

试题三 写作要点

软件单元测试、静态测试和动态测试，白盒测试的覆盖标准以及回归测试如何实施。

仅从单元测试来写一篇论文，内容较为狭隘，可能从单元测试的以下方面来写：

（1）测试的内容

模块接口测试、局部数据结构测试、边界条件测试、路径覆盖测试、错误处理与异常测试。

（2）测试的执行场景

描述为什么要进行这些测试，以及测试的必要性。

撰写关于“软件单元测试的架构设计师”这一主题的论文，可以从介绍单元测试的基本概念入手，逐步深入到架构设计层面如何有效指导和实施单元测试策略，最终提出创新的设计方法或最佳实践案。

下面是简化的论文大纲及部分内容示例，仅供参考：

关键词：单元测试，架构设计，测试策略，软件质量，开发效率

1.单元测试基础理论

2.架构设计与测试的关联性

3.现有单元测试方法及局限性

4.研究缺口分析

基于架构设计的单元测试策略

架构视角下的单元测试原则

●模块独立性

●测试可达性

●依赖管理

设计阶段的单元测试考虑

●接口设计与测试先行

●模块边界清晰化

实施框架与工具选择

●自动化测试框架

●持续集成/持续部署(CI/CD)集成

实现步骤：

项目背景与目标系统概述，为什么需要进行单元测试

测试架构设计与实施步骤

识别核心模块与接口

设计测试用例与桩/模拟对象

●集成自动化测试流程

结果分析与评估

●测试覆盖率提升

●缺陷检测率与修复周期

●对开发流程的影响

结论：

●使用了单元测试给项目和系统带来哪些好处，收到了.....好评；

●总结与建议。

试题四 论云上自动化运维级其应用

云上自动化运维是传统 IT 运维和 DevOps 的延伸，通过云原生架构实现运维的再进化。云上自动化运维可以有效帮助企业降低 IT 运维成本，提升系统的灵活度，以及系统的交付速度，增强系统的可靠性，构建更加安全、可信、开放的业务平台。

请围绕“云上自动化运维及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与运维的软件项目以及你在其中所承担的主要工作。

2.请简要描述云上自动化运维(如 CloudOps)的主要衡量指标。

3.具体阐述你所参与的项目是如何进行云上自动化运维的。

试题四 写作要点

云原生 DevOps、云上运维是新技术。以下是云原生 DevOps 简介。

1.核心要素：

●容器化：利用 Docker 等技术，实现应用的轻量级打包与标准化部署。

●微服务架构：将应用拆分成小型、独立的服务，便于独立开发、部署和扩展。

●持续集成/持续部署(CI/CD)：自动化代码集成、测试与部署，加速软件交付流程。

●基础设施即代码(IaC)：使用代码(如 Terraform、Cloud Formation)管理基础设施配置。

●服务网格：如 Istio，管理服务间的交互，提供安全、监控与故障恢复等功能。

2.云上运维特点：

（1）优势

●弹性与可扩展性：根据需求自动调整资源，应对流量高峰。

●成本效益：按需付费模型，减少闲置资源成本。

●自动化运维：利用云服务商工具(如 AWS Cloud Formation、Azure Automation)实现运维任务自动化。

●全球部署：轻松实现应用的全球分布与低延迟访问。

●安全性与合规：利用云平台提供的安全服务和最佳实践，保障数据安全与合规性。

（2）实施策略

1.拥抱容器化与 Kubernetes：作为云原生基础，实现应用的高效部署与管理。

2.建立 CI/CD 管道：集成如 Jenkins、GitLabCI/CD，确保代码变更快速、可靠地交付到生产环境。

3.采用微服务设计：分解大型应用，提高开发效率与系统的可维护性。

4.实施全面监控与日志管理：利用 ELKStack、Prometheus+Grafana 等工具，实现系统状态的实时监控与问题快速定位。

5.强化安全性：集成云原生安全解决方案，如密钥管理、网络策略、安全扫描等。

6.文化和组织调整：推动 DevOps 文化，促进开发、运维及其它团队之间的紧密合作与知识共享。

（3）协同工作

云原生 DevOps 与云上运维的协同，关键在于实现开发与运维流程的高度自动化与协同，确保从代码提交到生产部署的每一个环节都能快速、安全、可靠。通过云原生技术栈的运用，结合云平台的强大功能，可以极大地提升软件交付的速度与质量，同时降低运维复杂度和成本。此外，持续的学习与反馈循环也是保持体系高效运行的重要组成部分。

4.2 2023 下半年

试题一 可靠性分析与评价方法

机试后无详细题目描述，但要注意考的不是设计，而是分析和评价，涉及的内容是第二版教材新增的可靠性章节里的分析模型和评价方法。

（可靠性模型大致可分为如下 10 类。

·种子法模型。Jelinski-Moranda 的几何 De-eutrophication 模型。

·失效率类模型。Schick-Wolverton 模型。

·曲线拟合类模型。改进的 Schick-Wolverton 模型。

·可靠性增长模型。Moranda 的几何泊松模型。

·程序结构分析模型。Goal 和 0kumoto 不完全排错模型。

·输入域分类模型。

·执行路径分析方法模型。

·非齐次泊松过程模型。并联模型和串联模型

·马尔可夫过程模型。

·贝叶斯分析模型。）

试题二 面向对象分析

用例模型、分析模型、UML 图

（用例模型：识别参与者、合并需求获得用例、细化用例描述（用例名称、简要说明、事件流、非功能需求、前置条件、后置条件、扩展点、优先级）、调整用例模型（包含关系、扩展关系、泛化关系）

分析模型：√定义概念类

√识别类之间的关系：

·依赖关系

·关联关系

·聚合关系

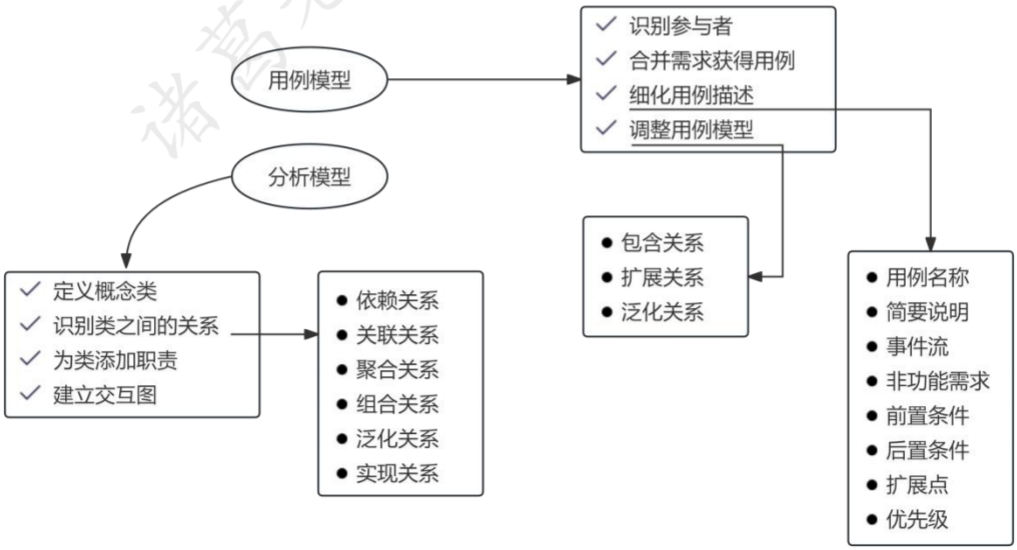
·组合关系

·泛化关系

·实现关系

√为类添加职责

√建立交互图）



试题三 多数据源

多源数据就是针对不同来源、不同格式、不同标准的数据，供企业进行数据共享，多源数据集成的技术主要有联邦式、基于中间件模型的以及数据仓库的。

1、简要分析当前的多源数据集成技术，分析他们的侧重点以及应用场景

2、描述你是怎么建立多源数据集成应用

（多源数据集成及其应用

题干里对多源数据集成做了简要介绍，另外题干里还顺带点了一下三种数据集成的方式(只是提到了名字)：数据仓库、联邦、还以一个忘了。介绍项目，介绍项目里怎样应用了多源数据集成，然后说明这样做的效果。

题目最下方 3 个问题里的第二个是：介绍几个常见的多源数据集成方法(没指定是哪几个)，并说明它们分别在哪种场景下适用。）

试题四 边云协同

边云协同：边缘计算与云计算各有所长，云计算擅长全局性、非实时、长周期的大数据处理与分析，能够在长周期维护、业务决策支撑等领域发挥优势：边缘计算更适用局部性、实时、短周期数据的处理与分析，能更好地支撑本地业务的实时智能化决策与执行。

边缘计算既靠近执行单元，更是云端所需高价值数据的采集和初步处理单元，可以更好地支撑云端应用：反之，云计算通过大数据分析优化输出的业务规则或模型可以下发到边缘侧，边缘计算基于新的业务规则或模型运行。

（主要包括六种协同：

（1）资源协同：边缘节点提供计算、存储、网络、虚拟化等基础设施资源、具有本地资源调度管理能力，同时可与云端协同，接受并执行云端资源调度管理策略，包括边缘节点的设备管理、资源管理以及网络连接管理。

（2）数据协同：边缘节点主要负责现场/终端数据的采集，按照规则或数据模型对数据进行初步处理与分析，并将处理结果以及相关数据上传给云端：云端提供海量数据的存储、分析与价值挖掘。边缘与云的数据协同，支持数据在边缘与云之间可控有序流动，形成完整的数据流转路径，高效低成本对数据进行生命周期管理与价值挖掘。

（3）智能协同：边缘节点按照 AI 模型执行推理，实现分布式智能；云端开展 AI 的集中式模型训练，并将模型下发边缘节点。

（4）应用管理协同：边缘节点提供应用部署与运行环境，并对本节点多个应用的生命周期进行管理调度：云端主要提供应用开发、测试环境，以及应用的生命周期管理能力。

（5）业务管理协同：边缘节点提供模块化、微服务化的应用/数字孪生/网络等应用实例：云端主要提供按照客户需求实现应用/数字孪生/网络等的业务编排能力。

（6）服务协同：边缘节点按照云端策略实现部分 ECSaaS 服务，通过 ECSaaS 与云端 SaaS 的协同实现面向客户的按需 SaaS 服务：云端主要提供 SaaS 服务在云端和边缘节点的服务分布策略，以及云端承担的 SaaS 服务能力。）

4.3 2022 下半年

试题一 论基于构件的软件开发方法及其应用

基于构作的软件开发(Component-Based Software Development，CBSD)是一种基于分布对象技术、强调通过可复用构件设计与构造软件系统的软件复用途径。基于构件的软件系统中的构件可以是 COTS（Commercial-Off-the-Shelf）构件，也可以是通过其它途径获得的构件(如自行开发)。CBSD 将软件开发的重点从程序编写转移到了基于已有构件的组装，以更快地构造系统，减轻用来支持和升级大型系统所需要的维护负担，从而降低软件开发的费用。

请围绕“基于构件的软件开发方法及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目，以及你在其中所承担的主要工作。

2.详细论述基于构件的软件开发方法的主要过程。

3.结合你具体参与管理和开发的实际项目，请说明具体实施过程以及碰到的主要问题。

试题一 写作要点

CBSD 方法使得软件开发不再一切从头开发，开发的过程就是构件组装的过程，维护的过程就是构件升级、替换和扩充的过程，其优点是提高了软件开发的效率；构件可由一方定义其规格说明，被另一方实现，然后供给第三方使用，CBSD 允许多个项目同时开发，降低了费用，提高了可维护性，可实现分步提交软件产品。

CBSD 方法由软件的需求分析和定义、架构设计、构件库的建立、应用软件构建、测试和发布五个阶段组成。

（1）需求分析和定义：在这阶段需要重点说明系统跟曾经开发过的其他系统类似，具有大量可复用的成熟构件。

（2）架构设计：结合实际项目，根据上一阶段获得的需求和定义提出架构模型。

（3）构件库的建立：这是本论文主题的重点。构件的获得有四个途径：

●从现有构件中获得符合要求的构件，直接使用或作适应性修改，得到可复用的构件。

●通过遗留工程（Legacy Engineering），将具有潜在复用价值的构件提取出来，得到可复用的构件。

●从市场上购买现成的商业构件，即 COTS（Commercial Off-The-Shell）构件。

●开发新的符合要求的构件。

而构件库的检索方法有 3 种：基于关键字的检索、刻面检索法、超文本检索法。

（4）应用软件构建：构建过程主要是构件的组装过程，而大致有三种技术：

●基于功能的组装技术。基于功能的组装技术采用子程序调用和参数传递的方式将构件组装起来。它要求库中的构件以子程序/过程/函数的形式出现，并且接口说明必须清晰。当使用这种组装技术进行软件开发时，开发人员首先要对新系统进行功能分解，将系统分解为强内聚、松耦合的功能模块；然后根据各模块的功能需求提取构件，进行适应性修改后，再挂接到上述功能分解框架中。

●基于数据的组装技术。基于数据的组装技术首先根据当前软件问题的核心数据结构设计出一个框架，然后根据框架中各结点的需求提取构件并进行适应性修改，再将构件逐个分配至框架中的适当位置。此后，构件的组装方式仍然是传统的子程序调用与参数传递。这种组装技术也要求库中构件以子程序形式出现，但它所依赖的软件设计方法不再是功能分解，而是面向数据的设计方法，例如，Jackson 系统开发方法。

●面向对象的组装技术。由于封装和继承特征，面向对象方法比其他软件开发方法更适合支持软件复用。在面向对象的软件开发方法中，如果从类库中检索出来的基类能够完全满足新系统的需求，则可以直接应用，否则，必须以基类为父类，生成相应的子类，以满足新系统的需求。

（5）测试和发布：可以是一个增量和迭代的过程。

试题二 论软件维护方法及其应用

软件维护是指在软件交付使用后，直至软件被淘汰的整个时间范围内，为了改正错误或满足新的需求而修改软件的活动。在软件系统运行过程中，软件需要维护的原因是多种多样的，根据维护的原因不同，可以将软件维护分为改正性维护、适应性维护、完善性维护和预防性维护。在维护的过程中，也需要对软件的可维护性进行度量。在软件外部，一般采用 MTTR 来度量软件的可维护性；在软件内部，可以通过度量软件的复杂性来间接度量软件的可维护性。

据统计，软件维护阶段占整个软件生命周期 60%以上的时间。因此，分析影响软件维护的因素，度量和提高软件的可维护性，就显得十分重要。

请围绕“软件维护方法及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目，以及你在其中所承担的主要工作。

2.详细论述影响软件维护工作的因素有哪些。

3.结合你具体参与管理和开发的实际项目，说明在具体维护过程中，如何度量软件的可维护性，说明具体的软件维护工作类型。

试题二 写作要点

影响软件的可维护性有以下 7 个因素。

1、可理解性。一个可维护的软件必然是可理解的。软件的可理解性是指通过阅读源代码和相关文档，了解软件的功能和如何运行的容易程度。软件的可理解性可以使用“90-10 测试”的方法来衡量，即如果一个有经验的程序员阅读一份源代码清单 10 分钟，可以写出该程序的 90%，则认为这个程序具有可理解性。

2、可测试性。一个可维护的软件必然是可测试的。软件的可测试性是指验证软件程序正确的难易程度。可测试性好的软件，通常意味着软件设计简单，复杂性低。因为软件的复杂性越大，测试的难度也就越大。

3、可修改性。一个可维护的软件必然是可修改的。软件的可修改性是指修改软件的难易程度。软件的可修改性可以通过进行几个简单的修改练习来评价。假设软件的平均复杂性是 C，要修改的模块的复杂性是 A，那么修改的难度可由下面公式计算：D=A/C。

4、可靠性。一个软件的可靠性越高，需要维护的概率就会越低。软件的可靠性是指软件在满足用户需求的前提下，在给定的时间段内正确运行的概率。软件可靠性的度量有以下两种方法：根据软件的错误统计进行可靠性预测。如度量软件的平均失效间隔时间（MTTF）。根据软件的复杂性进行可靠性预测。

5、可移植性。软件运行环境的变化是软件维护的一种常见情形，可移植性好的软件会降低维护的概率。软件的可移植性是指将软件从一个环境移植到新的的环境下正确运行的难易程度。一个可移植的软件应具有良好的结构，使用独立于机器的高级语言编写。

6、可使用性。软件易于使用通常意味着软件设计简单，易于理解。软件的可使用性是指用户使用软件的难易程度。软件的可使用性可以通过测试用户首次使用软件掌握常用功能的时间来衡量。

7、效率。效率是指软件既能很好地完成用户期望的功能、性能，又不浪费机器资源的程度。软件设计不能一味地追求效率，盲目地追求效率会使得软件的其它质量特性受到影响，比如降低软件的可维护性。

试题三 论区块链技术及应用

区块链作为一种分布式记账技术，目前已经被应用到了资产管理、物联网、医疗管理、政务监管等多个领域。从网络层面来讲，区块链是一个对等网络(Peer to Peer，P2P)，网络中的节点地位对等，每个节点都保存完整的账本数据，系统的运行不依赖中心化节点，因此避免了中心化带来的单点故障问题。同时，区块链作为一个拜占庭容错的分布式系统，在存在少量恶意节点情况下可以作为一个整体对外提供稳定的服务。

请围绕“区块链技术及应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及你在其中所承担的主要工作。

2.区块链包含多种核心技术，请简要描述区块链的 3 种核心技术。

3.具体阐述你参与管理和开发的项目是如何应用区块链技术进行设计与实现。

试题三 写作要点

区块链从本质上来看就是一个数据库，在其中存储的数据具备了“不可伪造，全程留痕，公开可追溯“等特性。区块链的四大核心技术如下：

1、分布式账本。指的是交易记账由分布在不同地方的多个节点共同完成，而且每一个节点记录的是完整的账目，因此它们都可以参与监督交易合法性，同时也可以共同为其作证。跟传统的分布式存储有所不同，区块链的分布式存储的独特性主要体现在两个方面：一是区块链每个节点都按照块链式结构存储完整的数据，传统分布式存储一般是将数据按照一定的规则分成多份进行存储。二是区块链每个节点存储都是独立的、地位等同的，依靠共识机制保证存储的一致性，而传统分布式存储一般是通过中心节点往其他备份节点同步数据。没有任何一个节点可以单独记录账本数据，从而避免了单一记账人被控制或者被贿赂而记假账的可能性。也由记账节点足够多，理论上讲除非所有的节点被破坏，否则账目就不会丢失，从而保证了账目数据的安全性。

2、非对称加密。存储在区块链上的交易信息是公开的，但是账户身份信息是高度加密的，只有在数据拥有者授权的情况下才能访问到，从而保证了数据的安全和个人的隐私。

3、共识机制。就是所有记账节点之间怎么达成共识，去认定一个记录的有效性，这既是认定的手段，也是防止篡改的手段。区块链提出了四种不同的共识机制，适用于不同的应用场景，在效率和安全性之间取得平衡。区块链的共识机制具备“少数服从多数”以及“人人平等”的特点，其中“少数服从多数”并不完全指节点个数，也可以是计算能力、股权数或者其他的计算机可以比较的特征量。“人人平等”是当节点满足条件时，所有节点都有权优先提出共识结果、直接被其他节点认同后并最后有可能成为最终共识结果。以比特币为例，采用的是工作量证明，只有在控制了全网超过 51%的记账节点的情况下，才有可能伪造出一条不存在的记录。当加入区块链的节点足够多的

时候，这基本上不可能，从而杜绝了造假的可能。

4、智能合约。是基于这些可信的不可篡改的数据，可以自动化的执行一些预先定义好的规则和条款。以保险为例，如果说每个人的信息（包括医疗信息和风险发生的信息）都是真实可信的，那就很容易的在一些标准化的保险产品中，去进行自动化的理赔。在保险公司的日常业务中，虽然交易不像银行和证券行业那样频繁，但是对可信数据的依赖是有增无减。

因此，笔者认为利用区块链技术，从数据管理的角度切入，能够有效地帮助保险公司提高风险管理能力。具体来讲主要分投保人风险管理和保险公司的风险监督。

试题四 论湖仓一体架构及其应用

随着 5G、大数据、人工智能、物联网等技术的不断成熟，各行各业的业务场景日益复杂，企业数据呈现出大规模、多样性的特点，特别是非结构化数据呈现出爆发式增长趋势。在这一背景下，企业数据管理不再局限于传统的结构化 OLTP(On-Line Transaction Processing)数据交易过程，而是提出了多样化、异质性数据的实时处理要求。传统的数据湖(Data Lake)在事务一致性及实时处理方面有所欠缺，而数据仓库(Data Warehouse)也无法应对高并发、多数据类型的处理。因此，支持事务一致性、提供高并发实时处理及分析能力的湖仓一体(Lake House)架构应运而生。湖仓一体架构在成本、灵活性、统一数据存储、多元数据分析等多方面具备优势，正逐步转化为下一代数据管理系统的核心竞争力。

请围绕“湖仓一体架构及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的、采用湖仓一体架构的软件项目以及你在其中所承担的主要工作。

2.请对湖仓一体架构进行总结与分析，给出其中四类关键特征，并简要对这四类关键特征的内涵进行阐述。

3.具体阐述你参与管理和开发的项目是如何采用湖仓一体架构的，并围绕上述四类关键特征，详细论述在项目设计与实现过程中遇到了哪些实际问题，是如何解决的。

试题四 写作要点

湖仓一体是一种新型的开放式架构，打通了数据仓库和数据湖，将数据仓库的高性能及管理能力与数据湖的灵活性融合了起来，底层支持多种数据类型并存，能实现数据间的相互共享，上层可以通过统一封装的接口进行访问，可同时支持实时查询和分析，为企业进行数据治理带来了更多的便利性。湖仓一体可在数据入湖后原地进行数据处理与分析，能有效避免数据冗余及流动导致的算力、网络及成本开销，可以作为超大型 ODS 存储贴源数据，实现全量数据的实时处理。

湖仓一体架构在数据管理中主要具有以下几大关键特征：

1、支持分析多种类型数据。湖仓一体架构可为多应用程序提供数据的入库、转换、分析和访问。数据类型包括结构化与非结构化类型，如文本、图像、视频、音频等，以及半结构化数据，如 JSON 等。

2、数据可治理，避免产生数据沼泽。湖仓一体架构可以支持各类数据模型的实现和转变，支持 DW 模式架构，例如星型模型、雪花模型等，可保证数据的完整性，同时具有健全的治理和审计机制，能够避免数据沼泽现象的出现。

3、事务支持。在企业中，数据库往往要为业务系统提供并发的数据读取和写入。湖仓一体架构对事务 ACID 的支持，可确保并发访问，尤其是 SQL 访问模式下的数据一致性、正确性。

4、BI 支持。湖仓一体支持直接在源数据上使用 BI 工具，这样可以提高分析效率，降低数据延时。另外，相比于在数据湖和数据仓库中分别操作两个副本的方式，湖仓一体更具成本优势。

5、存算分离。湖仓一体采用存算分离架构，可使系统能够扩展到更大规模的并发能力和数据容量，能满足新时代对于分布式数据架构的要求。

6、开放性。湖仓一体采用开放、标准化的存储格式（例如行存、列存、块存），能提供丰富的 API 支持。因此，各种工具和引擎（包括机器学习和 Python/R 库）可以高效地对数据进行直接访问。

从落地性来看，湖仓一体技术架构落地目前有三种方式：

1、基于 Hadoop 体系的数据湖向数据仓库能力扩展，湖中建仓，从数据湖进化到湖仓一体。湖仓一体结合了数据湖和数据仓库特点，直接在用于数据湖的低成本存储上实现与数据仓库中类似的数据结构和数据管理功能。目前主要有 Netflix 等开源企业在探索此技术路线。

2、基于自身云平台或第三方对象存储（如 OSS、S3、Ceph 等），基于 Hadoop 或自研技术进行湖仓一体能力的搭建。探索此技术路线的通常是各大云厂商，如 AWS、阿里云、华为云等。

3、以数据库技术为基础，自研分布式平台，从调度、计算到存储不依赖第三方平台，形成可以灵活在公有云、私有云、裸金属等场景独立部署使用的能力。技术方向上更注重于实时高并发场景及非结构化数据数据治理，并逐步向更广泛的分析场景发展，主要厂商以 Snowflakes、Databricks、巨杉数据库等为代表。

4.4 2021 下半年

试题一 论面向方面的编程技术及其应用（AOP）

面向过程编程是一种自顶向下的编程方法，其实质是对软件进行功能性分解。它适用于小型软

件系统，例如某一算法的实现。在大型应用系统中，自顶向下逐步求精的方法无论在系统体系结构的确立，系统的进化和维护，以及软件重用性方面都存在其不足之处。

请围绕“论面向方面的编程技术及其应用（AOP）”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

2.叙述在项目实践过程使用 AOP 技术开发的具体步骤。

3.结合项目，论述使用 AOP 的原因，开发过程中存在的问题及所使用技术带来的实际应用效果。

试题一 写作要点

面向方向编程（AOP）包括以下三个主要的步骤：

（1）方面分解。分解需求提取出横切关注点和核心关注点。把核心模块级关注点和系统级的横切关注点进行分离。例如，对于一个信用卡系统，可以分解出三个关注点：核心的信用卡处理、日志和验证。

（2）关注点实现。各自独立地实现这些关注点，用 OOP（面向对象的程序设计）实现核心关注点，用 AOP 实现横切关注点。例如，可以用 OOP 实现信用卡处理单元，而用 AOP 实现日志单元和验证单元。

（3）方面的重新组合。方面集成器通过创建一个模块单元（方面）来制定重组的规则，重组过程也称为编织。

AOP(Aspect-Oriented Programming，面向方面编程)，可以说是 OOP(Object-Oriented Programming，面向对象编程)的补充和完善。OOP 引入封装、继承和多态性等概念来建立一种对象层次结构，用以模拟公共行为的一个集合。当我们需要为分散的对象引入公共行为的时候，OOP 则显得无能为力。也就是说，OOP 允许你定义从上到下的关系，但并不适合定义从左到右的关系。例如日志功能，日志代码往往水平地散布在所有对象层次中，而与它所散布到的对象的核心功能毫无关系。对于其他类型的代码，如安全性、异常处理和透明的持续性也是如此。这种散布在各处的无关的代码被称为横切(cross-cutting)代码，在 OOP 设计中，它导致了大量代码的重复，而不利于各个模块的重用。

而 AOP 技术则恰恰相反，它利用一种称为“横切”的技术，剖解开封装的对象内部，并将那些影响了多个类的公共行为封装到一个可重用模块，并将其命名为"Aspect"，即方面。所谓“方面”，简单地说，就是将那些与业务无关，却为业务模块所共同调用的逻辑或责任封装起来，以减少系统的重复代码，降低模块间的耦合度，并有利于未来的可操作性和可维护性。AOP 代表的是一个横向的关系，如果说“对象”是一个空心的圆柱体，其中封装的是对象的属性和行为；那么面向方面编程的方法，就仿佛一把利刃，将这些空心圆柱体剖开，以获得其内部的消息。而剖开的切面，也就是所谓的“方面”了。然后它又以巧夺天工的妙手将这些剖开的切面复原，不留痕迹。

使用“横切”技术，AOP 把软件系统分为两个部分：核心关注点和横切关注点。业务处理的主要流程是核心关注点，与之关系不大的部分是横切关注点。横切关注点的一个特点是，他们经常发生在核心关注点的多处，而各处都基本相似，比如权限认证、日志、事务处理。AOP 的作用在于分离系统中的各种关注点，将核心关注点和横切关注点分离开来。

AOP 应用程序包括以下三个主要的开发步骤：

1.将系统需求进行功能性分解，区分出普通关注点以及横切关注点，确定哪些功能是组件语言必须实现的，哪些功能可以以 aspect 的形式动态加入到系统组件中。

2.单独完成每一个关注点的编码和实现，构造系统组件和系统 aspect。这里的系统组件，是实现该系统的基本模块，对于 OOP 语言，这些组件可以是类，对于过程化程序设计语言，这些组件可以是各种函数和 API。系统 aspect 是指用 AOP 语言实现的将横切关注点封装成的独立的模块单元。

3.用联接器指定的重组规则，将组件代码和 aspect 代码进行组合，形成最终系统。为达到此目的，应用程序需要利用或创造一种专门指定规则的语言，用它来组合不同应用程序片断。这种用来指定联结规则的语言可以是一种已有编程语言的扩展，也可以是一种完全不同的全新语言。

试题二 论系统安全架构设计及其应用

信息安全的特征是为了保证信息的机密性、完整性、可用性、可控性和不可抵赖性。

信息系统的安全保障是以风险和策略为基础，在信息系统的整个生命周期中提供包括技术、管理、人员和工程过程的整体安全，在信息系统中保障信息的这些安全特征，并实现组织机构的使命。许多信息系统的用户需要提供一种方法和内容对信息系统的技术框架、工程过程能力和管理能力提出安全性要求，并进行可比性的评估、设计和实施。

请围绕“论系统安全架构设计及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

2.详细论述安全架构设计中鉴别框架和访问控制框架设计的内容，并论述鉴别和访问控制所面临的主要威胁，并说明其危害。

3.阐述你在软件开发的过程中都遇到了哪些实际问题及解决方法。

试题二 写作要点

鉴别（Authentication）的基本目的，就是防止其他实体占用和独立操作被鉴别实体的身份。鉴别提供了实体声称其身份的保证，只有在主体和验证者的关系背景下，鉴别才是有意义的。鉴别有两种重要的关系背景：一是实体由申请者来代表，申请者与验证者之间存在着特定的通信关系（如实体鉴别）；二是实体为验证者提供数据项来源。

鉴别的方式主要基于以下 5 种。

（1）已知的，如一个秘密的口令。

（2）拥有的，如 1C 卡、令牌等。

（3）不改变的特性，如生物特征。

（4）相信可靠的第三方建立的鉴别（递推）。

（5）环境（如主机地址等）。

鉴别服务分为以下阶段：安装阶段；修改鉴别信息阶段；分发阶段；获取阶段；传送阶段；验证阶段；停活阶段；重新激活阶段；取消安装阶段。

在安装阶段，定义申请 AI 和验证 AI.修改鉴别信息阶段，实体或管理者申请 AI 和验证 AI 变更（如修改口令）。在分发阶段，为了验证交换 AI，把验证 AI 分发到各实体（如申请者或验证者）以供使用。在获取阶段，申请者或验证者可得到为鉴别实例生成特定交换 AI 所需的信息，通过与可信第三方进行交互或鉴别实体间的信息交换可得到交换 AI.例如，当使用联机密钥分配中心时，申请者或验证者可从密钥分配中心得到一些信息，如鉴别证书。在传送阶段，在申请者与验证者之间传送交换 AI.在验证阶段，用验证 AI 核对交换 AI。在停活阶段，将建立一种状态，使得以前能被鉴别的实体暂时不能被鉴别。在重新激活阶段，使在停活阶段建立的状态将被终止。在取消安装阶段，实体从实体集合中被拆除。

访问控制（Access Control)决定开放系统环境中允许使用哪些资源、在什么地方适合阻止未授权访问的过程。在访问控制实例中，访问可以是对一个系统（即对一个系统通信部分的一个实体）或对一个系统内部进行的。

ACI(访问控制信息）是用于访问控制目的的任何信息，其中包括上下文信息。AD1(访问控制判决信息）是在做出一个特定的访问控制判决时可供 ADF 使用的部分（或全部）ACI。ADF(访问控制判决功能）是一种特定功能，它通过对访问请求、ADI 以及该访问请求的上下文使用访问控制策略规则而做出访问控制判决。AEF(访问控制实施功能）确保只有对目标允许的访问才由发起者执行。

涉及访问控制的有发起者、AEF、ADF 和目标。发起者代表访问或试图访问目标的人和基于计算机的实体。目标代表被试图访问或由发起者访问的，基于计算机或通信的实体。例如，目标可能是 OSI 实体、文件或者系统。访问请求代表构成试图访问部分的操作和操作数。

当发起者请求对目标进行特殊访问时，AEF 就通知 ADF 需要一个判决来做出决定。为了作出判决，给 ADF 提供了访问请求（作为判决请求的一部分）和下列几种访问控制判决信息（ADI）。

试题三 论企业集成平台的理解与应用

企业信息集成是解决“孤岛”问题的需要，技术发展的同时也推动了集成架构等相关的研究。企业集成平台的核心是企业集成架构，包括信息、过程、应用集成的架构。

请围绕“论企业集成平台的理解与应用”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

2.给出至少 4 种企业集成平台应具有的基本功能，并对这 4 种功能内涵进行简述。

3.阐述你在软件开发的过程中都遇到了哪些实际问题及解决方法。

试题三 写作要点

集成平台是支持企业集成的支撑环境，包括硬件、软件、软件工具和系统，通过集成各种企业应用软件形成企业集成系统。由于硬件环境和应用软件的多样性，企业信息系统的功能和环境都非常复杂，因此，为了能够较好地满足企业的应用需求，作为企业集成系统支持环境的集成平台，其基本功能要如下。

（1）通信服务

提供分布环境下透明的同步/异步通信服务功能，使用户和应用程序无需关心具体的操作系统和应用程序所处的网络物理位置，而以透明的函数调用或对象服务方式完成它们所需的通信服务要求。

（2）信息集成服务

为应用提供透明的信息访问服务，通过实现异种数据库系统之间数据的交换、互操作、分布数据管理和共享信息模型定义（或共享信息数据库的建立)，使集成平台上运行的应用、服务或用户端能够以一致的语义和接口实现对数据（数据库、数据文件、应用交互信息）的访问与控制。

（3）应用集成服务

通过高层应用编程接口来实现对相应应用程序的访问，这砦高层应用编程接口包含在不同的适配器或代理中，被用来连接不同的应用程序。这些接口以函数或对象服务的方式向平台的组件模型提供信息，使用户在无需对原有系统进行修改（不会影响原有系统的功能）的情况下，只要在原有系统的基础上加上相应的访问接口就可以将现有的、用不同的技术实现的系统互联起来，通过为应用提供数据交换和访问操作，使各种不同的系统能够相互协作。

（4）二次开发工具

是集成平台提供的一组帮助用户开发特定应用程序（如实现数据转换的适配器或应用封装服务等）的支持工具，其目的是简化用户在企业集成平台实施过程中（特定应用程序接口）的开发工作。

（5）平台运行管理工具

是企业集成平台的运行管理和控制模块，负责企业集成平台系统的静态和动态配置、集成平台应用运行管理和维护、事件管理和出错管理等。通过命名服务、目录服务、平台的动态静态配置，

以及其中的关键数据的定期备份等功能来维护整个服务平台的系统配置及稳定运行。

试题四 论微服务架构及其应用

微服务提倡将单一应用程序划分成一组小的服务，服务之间互相协调、互相配合，为用户提供最终价值。每个服务运行在其独立的进程中，服务与服务间采用轻量级的通信机制互相沟通。在微服务架构中，每个服务都是一个相对独立的个体，每个服务都可以选择适合于自身的技术来实现。每个服务的部署都是独立的，这样就可以更快地对特定部分的代码进行部署。

请围绕“论微服务架构及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

2.简要描述微服务优点。

3.具体阐述如何基于微服务架构进行软件设计实现的。

试题四 写作要点

微服务好处：高异构性，高性能，高弹性，高扩展，易部署，可组合性，可替代性

微服务优点：

●通过应用“分而治之”的原则，持续交付和部署大型，复杂的应用程序

●通过更易于理解，开发和测试系统来提高模块化

●通过每个微服务具有较小的代码库来降低复杂性

●允许更新功能，而对系统的其余部分没有影响或影响极小

●使架构变得高度可扩展

●大大减少了破坏系统无关部分的机会

●可以独立交付和部署服务，而不必等待整个系统发布

●允许部署到多个云和本地基础设施环境

●在持续发展现有系统的同时持续融入和利用最新的技术

●使同一时间在同一系统上工作的一组开发人员间的协作更可控

基于微服务的系统设计实现：

设计原则

●围绕业务概念建模

●实现自动化

●隐藏内部实现细节

●一切去中心化

●独立部署

●隔离失败

●高度可观察

微服务 RESTful API：业务服务及通用服务

服务网关 API Gateway：客户端到微服务通信

服务注册 Service Registry：微服务注册，发现中心

事件总线 Event Bus：微服务到微服务通信

安全保护 Auth Provider：认证授权提供服务

4.5 2020 下半年

试题一 论企业集成架构设计及应用

企业集成架构（Enterprise Integration Architecture，EIA）是企业集成平台的核心，也是解决企业信息孤岛问题的关键。企业集成架构设计包括了企业信息、业务过程、应用系统集成架构的设计。实现企业集成的技术多种多样，早期的集成方式是通过在不同的应用之间开发一对一的专用接口来实现应用之间的数据集成，即采用点到点的集成方式；后来提出了利用集成平台的方式来实现企业集成，可以将分散的信息系统通过一个统一的接口，以可管理、可重复的方式实现单点集成。企业集成架构设计技术方案按照要解决的问题类型可以分为数据集成、应用集成和企业集成。

请围绕“论企业集成架构设计及应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与的软件开发项目以及承担的主要工作。

2.详细说明三类企业集成架构设计技术分别要解决的问题及其含义，并阐述每种技术具体包含了哪些集成模式。

3.根据你所参与的项目，说明采用了哪些企业集成架构设计技术，其实施效果如何。

试题一写作要点

一、简要描述所参与的软件系统开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、详细说明三类企业集成架构设计技术分别要解决的问题及其含义，并阐述每种技术具体包含了哪些集成模式。

企业集成架构有三类，分别是数据集成、应用集成、企业集成。

●数据集成，是为了解决不同应用和系统间的数据共享和交换需求，具体包括共享信息管理、共享模型管理和数据操作管理三个部分。共享信息管理通过定义统一的集成服务模型和共享信息访问机制，完成对集成平台运行过程中产生数据信息的共享、分发和存储管理；共享模型管理则提供数据资源配置管理、集成资源关系管理、资源运行生命周期管理及相应的业务数据协同监控管理等功能；数据操作管理则为集成平台用户提供数据操作服务，包括多通道的异构模型之间的数据转换、数据映射、数据传递和数据操作等功能服务。数据集成的模式包括：数据联邦、数据复制模式、基于接口的数据集成模式。

●应用集成，是指两个或多个应用系统根据业务逻辑的需要而进行的功能之间的互相调用和互操作。应用集成需要在数据集成的基础上完成。应用集成在底层的网络集成和数据集成的基础上实现异构应用系统之间应用层次上的互操作。它们共同构成了实现企业集成化运行最顶层集成所需要的技术层次上的支持。应用集成的模式包括：集成适配器模式、集成信使模式、集成面板模式和集成代理模式。

●企业集成，应用软件系统从功能逻辑上可以分为表示、业务逻辑和数据三个层次。其中表示层负责完成系统与用户交互的接口定义；业务逻辑层主要根据具体业务规则完成相应业务数据的处理；数据层负责存储由业务逻辑层处理所产生的业务数据，它是系统中相对稳定的部分。支持企业间应用集成和交互的集成平台通常采用多层结构，其目的是在最大程度上提高系统的柔性。在集成平台的具体设计开发中，还需要按照功能的通用程度对系统实现模块进行分层。企业集成的模式包括：前端集成模式、后端集成模式和混合集成模式。

三、针对考生本人所参与的项目中使用的企业集成架构设计技术，说明实施过程和具体实施效果。

试题二 论软件测试中缺陷管理及其应用

软件缺陷指的是计算机软件或程序中存在的某种破坏正常运行能力的问题、错误，或者隐藏的功能缺陷。缺陷的存在会导致软件产品在某种程度上不能满足用户的需要。在目前的软件开发过程

中，缺陷是不可避免的。软件测试是发现缺陷的主要手段，其核心目标就是尽可能多地找出软件代码中存在的缺陷，进而保证软件质量。软件缺陷管理是软件质量管理的一个重要组成部分。

请围绕“论软件测试中缺陷管理及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及承担的主要工作。

2.详细论述常见的缺陷种类和级别，论述缺陷管理的基本流程。

3.结合你具体参与管理和开发的实际项目，说明是如何进行缺陷管理的，请说明具体实施过程以及应用效果。

试题二 写作要点

一、简要叙述所参与管理和开发的软件项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、根据 IEEE 标准，软件测试中所发现的缺陷主要包括：输入/输出错误；逻辑错误；计算错误；接口错误；数据错误等；从软件测试角度还可以将缺陷分为五类：功能缺陷；系统缺陷；加工缺陷；数据缺陷；代码缺陷。不同企业的缺陷分类往往不同。

根据缺陷后果的严重程度，可以将缺陷分为多个不同的级别，例如 Beizer 将缺陷分为十级：轻微、中等、使人不悦、影响使用、严重、非常严重、极为严重、无法容忍、灾难性、传染性等。

缺陷管理是对软件测试环节中缺陷状态的完整跟踪和管理，确保每个被发现的缺陷都得到妥善处理。

缺陷管理的目的是对各个阶段测试发现的缺陷进行跟踪管理，以保证各级缺陷的修复率达到标准，主要实现以下目标：保证信息的-致性；保证缺陷得到有效的跟踪；缩短沟通时间，解决问题更高效；收集缺陷数据并进行数据分析，作为缺陷度量的依据。

缺陷管理基本的流程如下：

（1）缺陷提交：测试人员发现缺陷后提交缺陷报告。

（2）缺陷审查：确定缺陷问题、种类和级别。

（3）修复流程：缺陷审查通过后进入修复流程，缺陷报告会转发给相应的软件开发人员进行修复。

（4）验证流程：开发人员提交修复后的代码，进入验证流程。通过回归测试等方法验证缺陷问题已经修复。

（5）缺陷关闭：在确认缺陷已完全解决后，关闭该缺陷。部分缺陷管理流程中，还包括对缺陷状态的跟踪。

三、考生需结合自身参与项目的实际状况，指出其参与管理和开发的项目中所进行的缺陷管理

活动，说明缺陷管理的具体实施过程，并对实际应用效果进行分析。

理论素材准备

1、软件缺陷定义

软件缺陷（Defect），常常又被叫做 Bug。所谓软件缺陷，即为计算机软件或程序中存在的某种破坏正常运行能力的问题、错误，或者隐藏的功能缺陷。缺陷的存在会导致软件产品在某种程度上不能满足用户的需要。IEEE729-1983 对缺陷有一个标准的定义：从产品内部看，缺陷是软件产品开发或维护过程中存在的错误、毛病等各种问题；从产品外部看，缺陷是系统所需要实现的某种功能的失效或违背。

2、产生原因

在软件开发的过程中，软件缺陷的产生是不可避免的。那么造成软件缺陷的主要原因有哪些？从软件本身、团队工作和技术问题等角度分析，就可以了解造成软件缺陷的主要因素。软件缺陷的产生主要是由软件产品的特点和开发过程决定的。

2.1 软件本身

●需求不清晰，导致设计目标偏离客户的需求，从而引起功能或产品特征上的缺陷。

●系统结构非常复杂，而又无法设计成一个很好的层次结构或组件结构，结果导致意想不到的问题或系统维护、扩充上的困难；即使设计成良好的面向对象的系统，由于对象、类太多，很难完成对各种对象、类相互作用的组合测试，而隐藏着一些参数传递、方法调用、对象状态变化等方面问题。

●对程序逻辑路径或数据范围的边界考虑不够周全，漏掉某些边界条件，造成容量或边系统界错误。

●对一些实时应用，要进行精心设计和技术处理，保证精确的时间同步，否则容易引起时间上不协调，不一致性带来的问题。

●没有考虑系统崩溃后的自我恢复或数据的异地备份、灾难性恢复等问题，从而存在系统安全性、可靠性的隐患。

●系统运行环境的复杂，不仅用户使用的计算机环境千变万化，包括用户的各种操作方式或各种不同的输入数据，容易引起一些特定用户环境下的问题；在系统实际应用中，数据量很大。从而会引起强度或负载问题。

●由于通信端口多、存取和加密手段的矛盾性等，会造成系统的安全性或适用性等问题。

.新技术的采用，可能涉及技术或系统兼容的问题，事先没有考虑到。

2.2 团队工作

●系统需求分析时对客户的需求理解不清楚，或者和用户的沟通存在一些困难。

●不同阶段的开发人员相互理解不一致。例如，软件设计人员对需求分析的理解有偏差，编程人员对系统设计规格说明书某些内容重视不够，或存在误解。

●对于设计或编程上的一些假定或依赖性，相关人员没有充分沟通。

●项目组成员技术水平参差不齐，新员工较多，或培训不够等原因也容易引起问题。

2.3 技术问题

●算法错误：在给定条件下没能给出正确或准确的结果。

●语法错误：对于编译性语言程序，编译器可以发现这类问题；但对于解释性语言程序，只能在测试运行时发现。

●计算和精度问题：计算的结果没有满足所需要的精度。

●系统结构不合理、算法选择不科学，造成系统性能低下。

●接口参数传递不匹配，导致模块集成出现问题。

2.4 项目管理的问题

●缺乏质量文化，不重视质量计划，对质量、资源、任务、成本等的平衡性把握不好，容易挤掉需求分析、评审、测试、等时间，遗留的缺陷会比较多。

●系统分析时对客户的需求不是十分清楚，或者和用户的沟通存在一些困难。

●开发周期短，需求分析、设计、编程、测试等各项工作不能完全按照定义好的流程来进行，工作不够充分，结果也就不完整、不准确，错误较多；周期短，还给各类开发人员造成太大的压力，引起一些人为的错误。

●开发流程不够完善，存在太多的随机性和缺乏严谨的内审或评审机制，容易产生问题。

●文档不完善，风险估计不足等。

试题三 论云原生架构及其应用

近年来，随看数字化转型不断深入，科技创新与业务发展不断融合，各行各业正在从大工业时代的固化范式进化成面向创新型组织与灵活型业务的崭新模式。在这一背景下，以容器和微服务架构为代表的云原生技术作为云计算服务的新模式，已经逐渐成为企业持续发展的主流选择。云原生架构是基于云原生技术的一组架构原则和设计模式的集合，旨在将云应用中的非业务代码部分进行最大化剥离，从而让云设施接管应用中原有的大量非功能特性（如弹性、韧性、安全、可观测性、灰度等），使业务不再有非功能性业务中断困扰的同时，具备轻量、敏捷、高度自动化的特点。云原生架构有利于各组织在公有云、私有云和混合云等新型动态环境中，构建和运行可弹性扩展的应用，其代表技术包括容器、服务网格、微服务、不可变基础设施和声明式 API 等。

请围绕“论云原生架构及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及承担的主要工作。

2.服务化、弹性、可观测、韧性和自动化是云原生架构重要的设计原则。请简要对这些设计原则的内涵进行阐述。

3.具体阐述你参与管理和开发的项目是如何采用云原生架构的，并围绕上述四类设计原则，详细论述在项目设计与实现过程中遇到了哪些实际问题，是如何解决的。

试题三 写作要点

一、简要叙述所参与管理和开发的软件项目，需要明确指出在其中承担的主要任务和开展的主要工作。

二、云原生架构的设计原则具体描述如下：

云原生架构设计原则有 7 条。

1、服务化原则

通过服务化架构拆分不同生命周期的业务单元，实现业务单元的独立迭代，从而加快整体的迭代速度，保证迭代的稳定性。同时，服务化架构采用的是面向接口编程方式，增加了软件的复用程度，增强了水平扩展的能力。服务化设计原则还强调在架构层面抽象化业务模块之间的关系，从而帮助业务模块实现基于服务流量（而非网络流量）的策略控制和治理，而无须关注这些服务是基于何种编程语言开发的。通过微服务，需要将单体应用进一步拆分，按业务边界重新划分成分布式应用.使应用与应用之间不再直接共享数据，而是通过约定好的契约进行通信，以提高扩展性.业务化垂直扩展(ScaleUp)，并将微服务数据水平扩展(ScaleOut)。

2、弹性原则

系统部署规模可以随着业务量变化自动调整大小，而无须根据事先的容量规划准备固定的硬件

和软件资源。优秀的弹性能力不仅能够改变企业的 IT 成本模式，使得企业不用再考虑额外的软硬件资源成本支出（闲置成本），也能更好地支持业务规模的爆发式扩张，不再因为软硬件资源储备不足而留下遗憾。简言之，弹性原则是指系统部署规模可以随着业务量变化自动调整大小，而无须根据事先的容量规划准备固定的硬件和软件资源。

3、可观测性原则

强调主动性，在云计算这样的分布式系统中，主动通过日志、链路跟踪和度量等手段，让一次 App 点击所产生的多次服务调用耗时、返回值和参数都清晰可见，甚至可以下钻到每次第三方软件调用、SQL 请求、节点拓扑、网络响应等信息中。运维、开发和业务人员通过这样的观测能力可以实时掌握软件的运行情况，并获得前所未有的关联分析能力，以便不断优化业务的健康度和用户体验。简言之，可观测性更强调主动性，在云计算这样的分布式系统中，主动通过日志，链路跟踪和度量等手段，让一次 App 点击所产生的多次服务调用耗时，返回值，参数都可见。

4、韧性原则

是指当软件所依赖的软硬件组件出现异常时，软件所表现出来的抵御能力。这些异常通常包括硬件故障、硬件资源瓶颈（如 CPU 或网卡带宽耗尽）、业务流量超出软件设计承受能力、影响机房正常工作的故障或灾难、所依赖软件发生故障等可能造成业务不可用的潜在影响因素。业务上线之后，在运行期的大部分时间里，可能还会遇到各种不确定性输入和不稳定依赖的情况。当这些非正常场景出现时，业务需要尽可能地保证服务质量，满足当前以联网服务为代表的“永远在线”的要求。因此，韧性能力的核心设计理念是面向失败设计，即考虑如何在各种依赖不正常的情况下，减小异常对系统及服务质量的影响并尽快恢复正常。简言之，韧性是指当软件所依赖的软硬件组件出现异常时，软件所表现出来的抵御能力。韧性原则的实践与常见架构主要包括:服务异步化能力，服务治理能力(重试/限流/降级/熔断/反压)、主从模式、集群模式、多 AZ(AvailabilityZone，可用区)的高可用、单元化、跨区域(Region)容灾、异地多活容灾等。

5、自动化原则

通过 IaC、GitOps、OAM、Operator 和大量自动化交付工具在 CI/CD(持续集成/持续交付)流水线中的实践，企业可以标准化企业内部的软件交付过程，也可以在标准化的基础上实现自动化，即通过配置数据自描述和面向终态的交付过程，实现整个软件交付和运维的自动。

6、零信任原则

传统安全架构认为防火墙内的一切都是安全的，而零信任模型假设防火墙边界已经被攻破，且每个请求都来自于不可信网络，因此每个请求都需要经过验证。第一不能基于 IP 配置安全策略；第二身份应该成为基础设施；第三标准的发布流水线。

7、架构持续演进原则

云原生架构本身也应该且必须具备持续演进的能力，而不是一个封闭式的，被设计后一成不变的架构。特别是在业务高速迭代，更应该考虑如何保证架构演进与业务发展之间的平衡。演进式架构是指软件开发的初始阶段，就可以通过可拓展和松耦合设计，让后续可能发生的变更更加容易，升级性重构的成本更低，并且能够发生在开发实践，发布实践和整体敏捷度等软件生命周期中的任何阶段。

三、论文中需要结合项目实际工作，详细论述在项目中是如何采用云原生架构进行系统的设计与实现的，并围绕云原生架构的设计原则，论述遇到了哪些实际问题，是采用何种方法解决的。

试题四 论数据分片技术及其应用

数据分片就是按照定的规则，将数据集划分成相互独立、正交的数据子集，然后将数据子集分布到不同的节点上。通过设计合理的数据分片规则，可将系统中的数据分布在不同的物理数据库中，达到提升应用系统数据处理速度的目的。

请围绕“论数据分片技术及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及承担的主要工作。

2.Hash 分片、一致性 Hash（Consistent Hash）分片和按照数据范围（Range Based）分片是三种常用的数据分片方式。请简要阐述三种分片方式的原理。

3.具体阐述你参与管理和开发的项目采用了哪些分片方式，并具体说明其实现过程和应用效果。

试题四 写作要点

一、简要叙述所参与管理和开发的软件项目，需要明确指出在其中承担的主要任务和开展的主要工作。

二、三种分片方式的具体描述如下：

数据分片技术作为目前架构设计中处理大数据的一种常规手段，当前被广泛用于缓存、数据库、消息队列等中间件的开发与使用当中。数据分片概念就是按照一定的规则，将数据集划分成相对独立的数据子集，然后将数据子集分布到不同的节点上，这个节点可以是逻辑上节点，也可以是物理上的节点。数据分片需要按照一定的规则，不同的分布式场景需要设计不同的规则，但基本都遵循同样的原则：按照最主要、最频繁使用的访问方式来分片。一般有以下三种方式对数据进行分片：hash 方式、一致性 hash、按照数据范围。

1、hash 方式

通过对数据（一般为 Key 值）先进行 hash 计算再取模的方式是一种简单且使用频繁的分片方式，

也就是 Hash(Key)%N，这里的 N 大部分情况下就是我们的结点个数，这种方式相对简单实用，一般场景下能够满足我们的要求。但 Hash 取模方式主要的问题是节点扩容或缩减的时候，会产生大量的数据迁移，比如从 N 台设备扩容到 N+1 台，绝大部分的数据都要在设备间进行迁移。该种方式代码实现较为简单，既可以采用 jdk 自带的 hash 方式也可以采用其他 hash 算法，大家可以自行搜索具体实现。

2、一致性 hash

一致性 hash 是将数据按照特征值映射到一个首尾相接的 hash 环上，同时也将节点映射到这个环上。对于数据，从数据在环上的位置开始，顺时针找到的第一个节点即为数据的存储节点。这种模式的优点在于节点一旦需要扩容或缩减的时候只会影响到 hash 环上相邻的节点，不会发生大规模的数据迁移。

常规的一致性 hash 分片模式也有缺点，一致性 hash 方式在增加节点的时候，只能分摊一个已存在节点的压力，在其中一个节点挂掉的时候，该节点的压力也会被全部转移到下一个节点。理想的目标是当节点动态发生变化时，已存在的所有节点都能参与进来，达到新的均衡状态。因此在实际开发中一般会引入虚拟节点（virtual node）的概念，即不是将物理节点映射在 hash 环上，而是将虚拟节点映射到 hash 环上。虚拟节点的数目远大于物理节点，因此一个物理节点需要负责多个虚拟节点的真实存储。操作数据的时候，先通过 hash 环找到对应的虚拟节点，再通过虚拟节点与物理节点的映射关系找到对应的物理节点。

引入虚拟节点后的一致性 hash 需要维护的元数据也会增加：第一，虚拟节点在 hash 环上的问题，且虚拟节点的数目又比较多；第二，虚拟节点与物理节点的映射关系。但带来的好处是明显的，当一个物理节点失效时，hash 环上多个虚拟节点失效，对应的压力也就会发散到多个其余的虚拟节点，事实上也就是多个其余的物理节点。在增加物理节点的时候同样如此。除此之外，可以根据物理节点的性能来调整每一个物理节点对于虚拟节点的数量，充分、合理利用资源。

3、按数据范围（range based）

按数据范围分片其实也就是基于数据的业务属性进行分片，如唯一编码、时间戳、使用频率等，比如在数据库层面按 ID 范围、按时间进行分库、分表、分片，按数据被访问频率分为热点库与历史库等方法，都是按数据范围方式的具体应用。基于数据范围的分片

模式需要贴合项目实际场景，使用中需要注意以下几点：

●分片与扩展实现比较简单，结合 ID 范围、时间结合业务自行实现即可；

●较为依赖备份机制，否则某个节点发生异常无法迅速恢复，可用性较难保证；

●对数据规模要有前瞻性的评估，例如按时间分片，需要考虑单位时间片内数据分布是否均匀；

●注意各分片数据之间的性能平衡，因为在常规场景下，无论采用哪种基于数据范围的分片模式，都是距离当前时间点较近的数据被访问和操作的几率较大，所以要特别注意随着数据规模与时间的推移，历史数据规模不断膨胀导致的整体性能下降。

每种分片方式是否适用，一方面需要结合项目的实际情况与规模，另一方面也要从几个常规的维度去评估：

●数据分片策略，也就是具体的分片方式；

●数据分片节点的动态扩展，随着数据量的逐步增长，是否能够通过增加节点来动态扩展适应；

●数据分片节点的负载均衡，结合分片策略能否保证数据均匀的分布在各个节点上以及各个节点的负载压力是否均衡；

●数据分片的可用性，当其中一个节点产生异常，能否将该节点的数据转移到其他节点上。

三、论文中需要结合项目实际工作，详细论述项目采用了哪些分片方式，并具体说明其实现过程和应用效果。

4.6 2019 下半年

试题一 论软件设计方法及其应用

软件设计（Software Design，SD）是根据软件需求规格说明书设计软件系统的整体结构、划分功能模块、确定每个模块的实现算法以及程序流程等，形成软件的具体设计方案。软件设计把许多事物和问题按不同的层次和角度进行抽象，将问题或事物进行模块化分解，以便更容易解决问题。分解得越细，模块数量也就越多，设计者需要考虑模块之间的耦合度。

请围绕“论软件设计方法及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

2.详细阐述有哪些不同的软件设计方法，并说明每种方法的适用场景。

3.详细说明你所参与的软件开发项目中，使用了哪种软件设计方法，具体实施效果如何。

试题一 写作要点

1．简要描述所参与设计的软件系统，并明确指出在系统设计过程种承担的主要工作。

2．分析系统设计的主要方法，并详细阐述每种设计方法。

1)净室方法

净室软件工程(净室方法)是软件开发的一种形式化方法，它可以生成高质量的软件。它使用盒结构规约进行分析和设计建模，并且强调将正确性验证(而不是测试)作为发现和消除错误的主要机制，使用统计的测试来获取认证被交付的软件的可靠性所必需的出错率信息。

净室方法从使用盒结构表示的分析和设计模型入手，一个“盒”在某特定的抽象层次上封装系统(或系统的某些方面)。通过逐步求精的过程，盒被精化为层次，其中每个盒具有引用透明性：每个盒规约的信息内容对定义其精华是足够的，不需要信赖于任何其他盒的实现。这使得分析人员能够层次地划分一个系统，从在顶层的本质表示转移向在底层的实现特定的细节。净室方法主要使用三种盒类型：黑盒、状态盒和清晰盒。

净室方法是一种严格的软件工程方法，它是一种强调正确性的数学验证和软件可靠性认证的软件过程模型，其目标和结果是非常低的出错率，这是使用非形式化方法难于或不可能达到的。

2) 结构化设计

结构化方法由结构化分析、结构化设计、结构化程序设计构成，它是一种面向数据流的开发方法。结构化分析是根据分解与抽象的原则，按照系统中数据处理的流程，用数据流图来建立系统的功能模型，从而完成需求分析工作。

①结构化设计原则

②结构化设计步骤

③结构化缺点

在结构化设计中，模块和模块之间的关系局限于信息流，限制了对模块之间众多关系的表达，也无法体现模块和模块之间其他的众多关系，包含各种各样的结构、行为、依赖、包含（在结构化设计中这种关系隐含在分层中）、继承、关联关系等等。结构化设计仅解决了模块在封装和信息隐藏方面的问题。

④结构化设计适合场景

3) 面向对象设计

面向对象的设计模型包含以包图表示的软件体系结构图、以交互图表示的用例实现图、完整精确的类图、针对复杂对象的状态图和用以描述流程化处理过程的活动图等。

①UML 与 4+1 视图

②设计原则

③设计模式

④面向对象设计适合场景

4) 原型法

结构化方法和面向对象方法有一个共同点：在系统开发初期必须明确系统的功能要求，确定系统边界。从工程学角度来看，这是十分自然的：解决问题之前必须明确要解决的问题是什么，然而对于信息系统建设而言，明确问题本身不是一件轻松的事情。

①原型分类

水平原型和垂直原型，抛弃原型、演进原型和递增原型

②原型类型的选择

③原型法适合场景

3．结合项目实践。针对实际参与的软件设计过程，说明所采用的设计方法，并描述其具体实施过程和效果。

试题二 论软件系统架构评估及其应用

对于软件系统，尤其是大规模复杂软件系统而言，软件系统架构对于确保最终系统的质量具有十分重要的意义。在系统架构设计结束后，为保证架构设计的合理性、完整性和针对性，保证系统质量，降低成本及投资风险，需要对设计好的系统架构进行评估。架构评估是软件开发过程中的重要环节。

请围绕“论软件系统架构评估及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

2.详细阐述有哪些不同的软件系统架构评估方法，并从评估目标、质量属性和评估活动等方面论述其区别。

3.详细说明你所参与的软件开发项目中，使用了哪种评估方法，具体实施过程和效果如何。

试题二 写作要点

1.简要描述所参与架构评估的软件系统，并明确指出在评估过程中承担的主要工作。

2.分析软件系统架构评估中所普遍关注的质量属性，并详细阐述每种质量属性的具体含义。

系统架构评估中普遍关注的质量属性包括：

1) 性能

性能是指系统的响应能力，即需要多长时间才能对某个事件做出响应，或者在某段事件内系统所能处理的事件个数。经常用单位事件内所处理事务的数量或系统完成某个事务处理所需的时间来对性能进行定量表示。

2) 可靠性

可靠性是软件系统在应用或者系统错误面前，在意外或者错误使用的情况下维持软件系统的功能特性的基本能力。

3) 可用性

可用性是系统能够正常运行的时间比例。经常用两次故障之间的时间长度或在出现故障时系统能够恢复正常的速度来表示。

4) 安全性

安全性是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。

5) 可修改性

可修改性是指能够快速地以较高的性能价格比对系统进行变更的能力，包括可维护性、可扩展性、结构重构、可移植性。

6) 功能性

功能性是系统所能完成所期望的工作的能力。一项任务的完成需要系统中许多或大多数构件的相互协作。

7) 可变性

可变性是指体系结构经扩充或变更而成为新体系结构的能力。

8) 互操作性

互操作性是指作为系统组成部分的软件不是独立存在的，经常与其他系统或自身环境相互作用。如程序和用其他编程语言编写的软件系统的交互作用就是互操作性的问题。

3.分析软件系统架构评估种现阶段主要评估方法

业界已开发出多种软件架构评估的方法，按基于的技术手段来看，可以分为三类：基于调查问卷或检查表的方式、基于场景的方式和基于度量的方式。

1) 基于调查问卷或检查表的方式：该方式的关键是要设计好问卷或检查表，它充分利用系统相关人员的经验和知识，获得对架构的评估。其缺点是在很大程度上依赖于评估人员的主观推断。

2) 基于场景的方式：基于场景的方式由 SEI 首先提出并应用在架构权衡分析法(Architecture Trade off Analysis Method，ATAM)和软件架构分析方法(Software Architecture Analysis Method，SAAM)中。它是通过分析软件架构对场景(也就是对系统的使用或修改活动)的支持程度，从而判断该架构对这一场景所代表的质量需求的满足程度。

（1）架构权衡分析方法 ATAM 是一种系统架构评估方法，主要在系统开发之前，针对性能、

可用性、安全性和可修改性等质量属性进行评价和折中。ATAM 可以分为 4 个主要的活动阶段，包括需求收集、架构视图描述、属性模型构造和分析、架构决策与折中，整个评估过程强调以属性作为架构评估的核心概念。

（2）SAAM 是最早形成文档并得到广泛应用的软件架构分析方法。SAAM 的主要输入是问题描述、需求说明和架构描述，其分析过程主要包括场景开发、架构描述、单个场景评估、场景交互和总体评估。

3) 基于度量的方式：制定一些定量值来度量架构，如代码行数等。要制定质量属性和度量结果之间的映射。

4.针对实际参与的软件系统架构评估，说明所采用的评估方法，并描述其具体实施过程和效果。

试题三 论数据湖技术及其应用

近年来，随着移动互联网、物联网、工业互联网等技术的不断发展，企业级应用面临的数据规模不断增大，数据类型异常复杂。针对这一问题，业界提出“数据湖（Data Lake）”这一新型的企业数据管理技术。数据湖是一个存储企业各种原始数据的大型仓库，支持对任意规模的结构化、半结构化和非结构化数据进行集中式存储，数据按照原有结构进行存储，无须进行结构化处理；数据湖中的数据可供存取、处理、分析及传输，支撑大数据处理、实时分析、机器学习、数据可视化等多种应用，最终支持企业的智能决策过程。

请围绕“数据湖技术及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

2.详细阐述数据湖技术，并从主要数据来源、数据模式（Schema）转换时机、数据存储成本、数据质量、面对用户和主要支撑应用类型等方面详细论述数据湖技术与数据仓库技术的差异。

3.详细说明你所参与的软件开发项目中，如何采用数据湖技术进行企业数据管理，并说明具体实施过程以及应用效果。

试题三 写作要点

1．简要叙述所参与的大数据或数据湖系统项目，并明确指出在其中承担的职务和主要工作。

2．分析并阐述数据湖技术：

1)大数据的由来和表现。

2)大数据 5V 特性

3)大数据与数据仓库

4)数据仓库的 4 个特性

3.分析并阐述数据湖技术：

1)数据仓库的缺点

2)数据湖的技术特点

3)数据湖的概念

4)数据库的特性

4.分析并对比数据湖(Data Lake，DL)与数据仓库(Data Warehouse，DW)

1)数据来源

DW：从外部数据源获取

DL：既可从外部数据源获取，也可从外部应用或外部存储获取

2) 数据模式（Schema）转换时机

DW：采用 ETL

DL：既支持 ETL，也支持 ELT

3) 数据存储成本

DW：通常采用 ODS-DWD-DW-DM-ST 五层架构，存储成本高

DL：通常采用原始数据和流计算，存储成本低

4) 数据质量

DW：依赖数据治理规范，ELT 过程损失数据准确度和精度，质量较低

DL：存储原始数据，不损失数据准确度和精度，质量高

5) 面对用户和主要支撑应用类型

DW：联机分析(OLAP)和决策支持(DSS)

DL：物联网、人工智能和以智能制造、智慧城市等为代表的智能赋能应用

5．结合自身参与项目的实际状况，阐述数据湖技术在实际项目中的实施过程和应用效果。

试题四 论负载均衡技术在 Web 系统中的应用

负载均衡技术是提升 Web 系统性能的重要方法。利用负载均衡技术，可将负载（工作任务）进行平衡、分摊到多个操作单元上执行，从而协同完成工作任务，达到提升 Web 系统性能的目的。

请围绕“论负载均衡技术在 web 系统中的应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目，以及你在其中所承担的主要工作。

2.详细阐述常见的三种负载均衡算法，说明算法的基本原理。

3.详细说明你所参与的软件开发项目中，如何基于负载均衡算法实现 Web 应用系统的负载均衡。

试题四 写作要点

1．简要叙述所参与管理和开发的 Web 系统项目，并明确指出在其中承担的主要任务和开展的主要工作。

2．分析并描述负载均衡在 Web 应用项目高并发、高性能、高可用三高架构中的意义和作用

3.选择三种常见的通用负载均衡算法，详细阐述负载均衡中的算法和每种算法的基本原理：

1)轮询算法

2)随机算法

3)比率算法

4)优先级算法

5)最少连接数算法

6)最快响应时间算法

4．结合实际项目，详细阐述实际参与项目中采用了哪个或哪几个负载均衡算法实现，说明如何在项目中实践应用的负载均衡，并阐述负载均衡技术在项目中的应用效果。

4.7 2018 下半年

试题一 论软件开发过程 RUP 及其应用

RUP(Rational Unified Process，RUP)是 IBM 公司一款软件开发过程产品，它提出了一整套以 UML 为基础的开发准则，用以指导软件开发人员以 UML 为基础进行软件开发。RUP 汲取了各种面向对象分析与设计方法的精华，提供了一个普遍的软件过程框架，可以适应不同的软件系统、应用领域、组织类型和项目规模。

请围绕“论软件开发过程 RUP 及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及你在其中所担任的主要工作。

2.详细论述软件开发过程产品 RUP 所包含的 4 个阶段以及 RUP 的基本特征。

3.结合你所参与管理和开发的软件项目，详细阐述 RUP 在该项目中的具体实施内容，包括核心工作流的选择、制品的确定、各个阶段之间的演进及迭代计划以及工作流内部结构的规划等。

试题一 写作要点

一、简单介绍所参与的软件开发项目的背景及主要内容，说明在其中所担任的主要工作。

二、RUP 的 4 个阶段：初始阶段，定义最终产品视图和业务模型，并确定系统范围；细化阶段，设计及确定系统的体系结构，制定工作计划及资源要求；构造阶段，构造产品并继续演进需求、体

系结构、计划直至产品提交；移交阶段，把产品提交给用户使用。

RUP 的基本特征：受控的迭代式增量开发、用例驱动、以体系结构为中心。

1.受控的迭代式增量开发

（1）将软件开发分为一系列小的迭代过程，在每个迭代过程中逐步增加信息、进行细化；

（2）根据具体情况决定迭代的次数、每次迭代的持续时间以及迭代工作流；

（3）每次迭代都选择目前对风险影响最大的用例进行，以分解和降低风险。

2.用例驱动

（1）采用用例来捕获对目标系统的功能需求；

（2）采用用例来驱动软件的整个开发过程，保证需求的可追踪性，确保系统所有功能均被实现；

（3）将用户关心的软件系统的业务功能模型和开发人员关心的目标软件系统的功能实体模型结合起来，提供一种贯穿整个软件生存周期的开发方法，使得软件开发的各个阶段的工作自然、一致地协调起来。

3.以软件体系结构为中心

（1）强调在开发过程的早期，识别出与软件体系结构密切相关的用例，并通过对这些用例的分析、设计、实现和测试，形成体系结构框架；

（2）在后续阶段中对已经形成的体系结构框架进行不断细化，最终实现整个系统；

（3）在开发过程的早期形成良好的软件体系结构，有利于对系统的理解、支持重用和有效的组织软件开发。

三、结合具体项目，从以下 5 个方面具体说明 RUP 的具体实施内容。

1.确定本项目的软件开发过程需要哪些工作流。RUP 的 9 个核心工作流并不总是需要的，可以根据项目的规模、类型等对核心工作流做一些取舍。

2.确定每个工作流要产出哪些制品。

3.确定 4 个阶段之间如何演进。确定阶段间演进要以风险控制为原则，决定每个阶段要执行哪些工作流，每个工作流执行到什么程度，产出的制品有哪些，每个制品完成到什么程度等。

4.确定每个阶段内的迭代计划。规划 RUP 的 4 个阶段中每次迭代开发的内容有哪些。

5.规划工作流内部结构。工作流不是活动的简单堆积，工作流涉及角色、活动和制品，工作流的复杂程度与项目规模及角色多少等有很大关系。工作流的内部结构通常用活动图的形式给出。

试题二 论软件体系结构的演化

软件体系结构的演化是在构件开发过程中或软件开发完毕投入运行后，由于用户需求发生变化，

就必须相应地修改原有软件体系结构，以满足新的变化了的软件需求的过程。体系结构的演化是一个复杂的、难以管理的问题。

请围绕“论软件体系结构的演化”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及你在其中所承担的主要工作。

2.软件体系结构的演化是使用系统演化步骤去修改系统，以满足新的需求。简要论述系统演化的 6 个步骤。

3.具体阐述你参与管理和开发的项目是如何基于系统演化的 6 个步骤完成软件体系结构演化的。

试题二 写作要点

一、简要叙述所参与管理和开发的软件项目，需要明确指出在其中承担的主要任务和开展的主要工作。

二、软件体系结构的演化过程一般可分为以下 6 个步骤：

（1）需求变化归类

首先必须对用户需求的变化进行归类，使变化的需求与已有构件对应。对找不到对应构件的变动，也要做好标记，在后续工作中，将创建新的构件，以应对这部分变化的需求。

（2）制订体系结构演化计划

在改变原有结构之前，开发组织必须制订一个周密的体系结构演化计划，作为后续演化开发工作的指南。

（3）修改、增加或删除构件

在演化计划的基础上，开发人员可根据在第一步得到的需求变动的归类情况，决定是否修改或删除存在的构件、增加新构件。最后，对修改和增加的构件进行功能性测试。

（4）更新构件的互相作用

随着构件的增加、删除和修改，构件之间的控制流必须得到更新。

（5）构件组装与测试

通过组装支持工具把这些构件的实现体组装起来，完成整个软件系统的连接与合成，形成新的体系结构。然后对组装后的系统整体功能和性能进行测试。

（6）技术评审

对以上步骤进行确认，进行技术评审。评审组装后的体系结构是否反映需求变动，符合用户需求。如果不符合，则需要在第二步到第六步之间进行迭代。原来系统上所作的所有修改必须集成到原来的体系结构中，完成一次演化过程。

三、论文中需要结合项目实际工作，详细论述在项目中是如何基于上述系统演化的 6 个步骤实

现体系结构的演化的。

试题三 论面向服务架构设计及其应用

面向服务架构(Service-Oriented Architecture，SOA)是一种应用框架，将日常的业务应用划分为单独的业务功能服务和流程，通过采用良好定义的接口和标准协议将这些服务关联起来。通过实施甚于 SOA 的系统架构，用户可以构建、部署和整合服务，无需依赖应用程序及其运行平台，从而提高业务流程的灵活性，帮助企业加快发展速度，降低企业开发成本，改善企业业务流程的组织和资产重用。

请围绕“论面向服务架构设计及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与分析和开发的软件系统开发项目以及你所担任的主要工作。

2.说明面向服务架构的主要技术和标准，详细阐述每种技术和标准的具体内容。

3.详细说明你所参与的软件系统开发项目中，构建 SOA 架构时遇到了哪些问题，具体实施效果如何。

试题三 写作要点

一、简要描述所参与分析和开发的软件系统开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、说明面向服务架构的主要技术和标准，详细阐述每种技术和标准的具体内容。面向服务架构的主要技术和标准包括：

（1）UDDI(统一描述、发现和集成协议)

UDDI 实现了商业实体的发布、查找和发现机制，它定义了商业实体之间在网络上互相作用和共享信息。通过构建 UDDI 模块，使得商业实体能够快读、方便地使用它们自身的企业应用软件来发现合适的商业对等实体，并与其实施电子化的商业贸易。UDDI 中包含了服务描述与发现的标准规范。

（2）WSDL(Web 服务描述语言)

WSDL 是一个用来描述 Web 服务和说明如何与 Web 服务通信的 XML 语言。它是 Web 服务的接口定义语言，通过 WSDL，可以描述 Web 服务的三个基本属性：包括服务所提供的操作和服务交互的数据格式及协议、协议地址等信息。WSDL 以端口集合的形式来描述服务，包含了对一组操作和消息的抽象定义，绑定到这些操作和消息的一个具体协议，和这个绑定的一个网络端点规范。WSDL 分为服务接口描述恶化服务实现描述两种类型。

（3）SOAP(简单对象访问协议)

SOAP 是在分散或者分布式环境中基于 XML 的信息交换协议。SOAP 中包含了四个主要部分：SOA 封装定义了一个描述消息中的内容是什么，是谁发送的，谁应当接受并处理它以及如何处理它们的框架；SOAP 编码规则用于表示应用程序需要使用的数据类型的实例；SOAPRPC 表示约定了远程过程调用和应答的协议；SOAP 绑定使用底层协议交换信息。

（4）BPEL(业务流程执行语言)

BPEL 是面向 Web 服务的服务定义和执行过程描述的语言，用户可以通过组合、编排和协调 Web 服务自上而下的实现面向服务的体系结构。BPEL 提供了一种相对简单易懂的方法，可以将多个 Web 服务按照业务流程组合到一个新的组合服务中，新的组合服务可以以一个新的 Web 服务方式被访问或者被组合成更大的服务。

三、针对考生实际参与的软件系统开发项目，说明构建 SOA 架构时遇到了哪些问题，并描述实施 SOA 后的实际应用效果。

主要问题可以分为三类：

（1）SOA 系统如何与原有系统中的功能进行集成；

（2）SOA 系统服务的设计以及服务粒度的控制；

（3）无状态服务的设计以及服务流程的组织。

试题四 论 NoSQL 数据库技术及其应用

随着互联网 Web2.0 网站的兴起，传统关系数据库在应对 Web2.0 网站，特别是超大规模和高并发的 Web2.0 纯动态 SNS 网站上已经显得力不从心，暴露了很多难以克服的问题，而非关系型的数据库则由于其本身的特点得到了非常迅速的发展。NoSQL(Not only SQL)的产生就是为了解决大规模数据集合及多种数据类型带来的挑战，尤其是大数据应用难题。目前 NoSQL 数据库并没有一个统一的架构，根据其所采用的数据模型可以分为 4 类：键值(Key-Value)存储数据库、列存储数据库、文档型数据库和图(Graph)数据库。

请围绕“NoSQL 数据库技术及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及你在其中所担任的主要工作。

2.详细论述常见的 NoSQL 数据库技术及其所包含的主要内容，并说明 NoSQL 数据库的主要适用场景。

3.结合你具体参与管理和开发的实际项目，说明具体采用哪种 NoSQL 数据库技术，并说明架构设计过程及其应用效果。

试题四 写作要点

一、简要叙述所参与管理和开发的软件项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、目前常见的 NoSQL 数据库主要分为 4 类。

1.键值(Key-Value)存储数据库：数据库主要会使用到一个哈希表，这个表中有一个特定的键和一个指针指向特定的数据。Key/value 模型对于 IT 系统来说的优势在于简单、易部署。但是如果 DBA 只对部分值进行查询或更新的时候，Key/value 就显得效率低下了。举例如：Tokyo Cabinet/Tyrant、Redis、Voldemort、Oracle BDB。

2.列存储数据库：该数据库通常是用来应对分布式存储的海量数据。键仍然存在，但是它们的特点是指向了多个列。这些列是由列家族来安排的。如：Cassandra，Hbase，Riak。

3.文档型数据库：该数据模型是版本化的文档，半结构化的文档以特定的格式存储，例如 JSON。文档型数据库可以看作是键值数据库的升级版，允许之间嵌套键值。而且文档型数据库比键值数据库的查询效率更高。如：CouchDB、MongoDb、SequoiaDB。

4.图(Graph)数据库：该数据库使用图模型，并且能够扩展到多个服务器上。NoSQL 数据库没有标准的查询语言(SQL)，因此进行数据库查询需要指定数据模型。许多 NoSQL 数据库都有 REST 式的数据接口或者查询 API。如：Neo4J、InfoGrid、InfiniteGraph。NoSQL 数据库在以下的这几种情况下比较适用：

1.数据模型比较简单；

2.需要灵活性更强的 IT 系统；

3.对数据库性能要求较高；

4.不需要高度的数据一致性；

5.对于给定 key，比较容易映射复杂值的环境。

三、考生需结合自身参与项目的实际状况，指出其参与管理和开发的项目中所进行的具体的 NoSQL 数据库设计，说明具体的架构设计过程、使用的方法和工具，并对实际应用效果进行分析。

4.8 2017 下半年

试题一 论软件系统建模方法及其应用

软件系统建模(Software System Modeling)是软件开发中的重要环节，通过构建软件系统模型可以帮助系统开发人员理解系统、抽取业务过程和管理系统的复杂性，也可以方便各类人员之间的交流。软件系统建模是在系统需求分析和系统实现之间架起的一座桥梁，系统开发人员按照软件系统模型

开发出符合设计目标的软件系统，并基于该模型进行软件的维护和改进。

请围绕"论软件系统建模方法及其应用"论题，依次从以下三个方面进行论述。

1.概要叙述你参与的软件系统开发项目以及你所担任的主要工作。

2.说明软件系统开发中常用的建模方法有哪几类?阐述每种方法的特点及其适用范围。

3.详细说明你所参与的软件系统开发项目中，采用了哪些软件系统建模方法，具体实施效果如何。

试题一 写作要点

一、简要描述所参与分析和开发的软件系统开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、说明软件系统开发中常用的建模方法有哪几类?阐述每种方法的特点及其适用范围。

软件系统开发中常用的建模方法包括：

（1）功能分解法

功能分解法以系统需要提供的功能为中心来组织系统。首先定义各种大的功能，然后把功能分解为子功能，同时定义功能间的接口。比较大的子功能还可以被进一步分解，直到我们可以对它进行明确的定义。总的思想就是将系统根据功能分而治之，然后根据功能的需求设计数据结构。

（2）数据流法/结构化分析建模方法

基本方法是跟踪系统的数据流，研究问题域中数据如何流动以及在各个环节上进行何种处理，从而发现数据流和加工。然后将问题域映射为数据流、加工以及数据存储等元素并组成数据流图，用加工和数据字典对数据流及其处理过程进行描述。

（3）信息工程建模法

在实体关系图基础上发展而来，其核心是构识别实体及其关系。实体用于描述问题域中的一个事物，它包含一组描述事物数据信息的属性；关系描述问题域中的各个事物之间在数据方面的联系，它可以带有自己的属性。发展之后的方法把实体叫作对象，把关系的属性组织到关系对象中，具有面向对象的某些特征

（4）面向对象建模法

从面向对象设计领域发展而来，它通过对象对问题域进行完整的映射，对象包括了事物的数据属性和行为特征；它用结构和连接如实反映问题域中事物之间的关系，比如分类、组装等；它通过封装、继承和消息机制等使问题域的复杂性得到控制。

三、针对作者实际参与的软件系统开发项目，说明所采用的系统建模方法，并描述这些建模方法所产生的实际应用效果。

试题二 论软件架构风格

软件体系结构风格是描述某一特定应用领域中系统组织方式的惯用模式。体系结构风格定义一个系统家族，即一个体系结构定义一个词汇表和一组约束。词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的。体系结构风格反应了领域中众多系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。

请围绕"论软件架构风格"论题，依次从以下三个方面进行论述。

1.概要叙述你参与分析和设计的软件系统开发项目以及你所担任的主要工作。

2.软件系统开发中常用的软件架构风格有哪些?详细阐述每种风格的具体含义。

3.详细说明你所参与分析和设计的软件系统是采用什么软件架构风格的，并分析采用该架构风格设计的原因。

试题二 写作要点

一、简要叙述所参与分析和开发的软件系统，并明确指出在其中承担的主要任务和开展的主要工作。

二、软件系统开发中常用的软件构架风格包括：

（1）管道/过滤器

在管道/过滤器风格的软件体系结构中，每个构件都有一组输入和输出，构件读输入的数据流，经过内部处理，然后产生输出数据流。

（2）数据抽象和面向对象

这种风格建立在数据抽象和面向对象的基础上，数据的表示方法和它们的相应操作封装在一个抽象数据类型或对象中。

（3）基于事件的隐式调用

基于事件的隐式调用风格的思想是构件不直接调用一个过程，而是触发或广播一个或多个事件。系统中的其他构件中的过程在一个或多个事件中注册，当一个事件被触发，系统自动调用在这个事件中注册的所有过程，这样，一个事件的触发就导致了另一个模块中的过程的调用。基于事件的隐式调用风格的主要特点是事件的触发者并不知道哪些构件会被这些事件影响。

（4）分层系统

层次系统组成一个层次结构，每一层为上层服务，并作为下层客户。

（5）仓库系统及知识库

在仓库风格中，有两种不同的构件：中央数据结构说明当前状态，独立构件在中央数据存储上

执行。若构件控制共享数据，则仓库是一传统型数据库。若中央数据结构的当前状态触发进程执行的选择，则仓库是一黑板系统。黑板系统：主要由三部分组成：①知识源。知识源中包含独立的、与应用程序相关的知识，知识源之间不直接进行通信；它们之间的交互只通过黑板来完成；②黑板数据结构：黑板数据是按照与应用程序相关的层次来组织的解决问题的数据，知识源通过不断地改变黑板数据来解决问题；③控制：控制完全由黑板的状态驱动，黑板状态的改变决定使用的特定知识。

（6）C2 风格

C2 体系结构风格可以概括为，通过连接件绑定在一起按照一组规则运作的并行构件网络。C2 风格中的系统组织规则如下：系统中的构件和连接件都有一个顶部和一个底部；构件的顶部应连接到某连接件的底部，构件的底部则应连接到某连接件的顶部，而构件与构件之间的直接连接是不允许的；一个连接件可以和任意数目的其他构件和连接件连接；当两个连接件进行直接连接时，必须由其中一个的底部到另一个的顶部。

（7）客户/服务器风格

C/S 体系结构有三个主要组成部分：数据库服务器、客户应用程序和网络。

（8）三层 C/S 结构风格

二层 C/S 结构是单一服务器且以局域网为中心的，所以难以扩展至大型企业广域网或 Internet 软、硬件的组合及集成能力有限，客户机的负荷太重，难以管理大量的客户机，系统的性能容易变坏，数据安全性不好。三层 C/S 体系结构是讲应用功能分成表示层、功能层和数据层三个部分，削弱二层 C/S 结构的局限性。

（9）浏览器/服务器风格

浏览器/服务器风格就是三层 C/S 结构的一种实现方式，具体结构为浏览器/Web 服务器/数据库服务器。

三、考生需要结合自身具体参与分析和开发的实际软件系统，说明在该系统的设计和实现中，采用了哪一种或多种软件架构风格，并分析采用这种软件架构风格设计的原因。

试题三 论无服务器架构及其应用

近年来，随着信息技术的迅猛发展和应用需求的快速更迭，传统的多层企业应用系统架构面临越来越多的挑战，已经难以适应这种变化。在这一背景下，无服务器架构(Serverless Architecture)逐渐流行，它强调业务逻辑由事件触发，具有短暂的生命周期，运行于无状态的轻量级容器中，并且由第三方代为管理。采用无服务器架构，业务逻辑以功能即服务(Function As a Service，FAAS)的

方式形成多个相互独立的功能组件，以标准接口的形式向外提供服务；同时，不同功能组件间的逻辑组织代码将存储在通用的基础设施管理平台中，业务代码仅在调用时才激活运行，当响应结束后占用的资源便会释放。

请围绕"无服务器架构及其应用’论题，依次从以下三个方面进行论述。

1.概要叙述你参与分析和设计的软件系统开发项目以及你所担任的主要工作。

2.与传统的企业应用系统相比较，基于无服务器架构的应用系统具有哪些特点，请例举至少 3 个特点，并进行解释。

结合你具体参与分析和设计的软件开发项目，描述该软件的架构，说明该架构是如何是如何采用无服务器架构模式的.，并说明在采用无服务器架构后软件开发过程中遇到的实际问题和解决方案。

试题三 写作要点

一、叙述你参与分析和开发的软件系统开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、与传统的企业应用系统相比，基于无服务器架构的应用具有如下特点：

从功能角度看，基于无服务器架构的应用系统只需要关注业务逻辑实现代码，无须关心承载这些代码的应用服务器如何部署。代码的部署和运维由第三方基础设施管理平台完成。

从开发角度看，基于无服务器架构的应用系统不需要考虑特定框架或开发库，从编程语言和环境的角度看更像是一个普通应用。基础设施管理平台负责解释和运行各种语言编写的代码，提供各种异构的运行环境。

从部署的角度看，基于无服务器架构的应用系统无须考虑如何部署业务代码，仅需要上传业务代码至基础设施管理平台，由管理平台自动进行服务器选择与代码部署。从运行和扩展的角度看，基于无服务器架构的应用系统业务逻辑运行在无状态的容器中，能够实现弹性、自动的水平扩展。系统开发者仅需要提供基本的并发业务处理功能，当系统面临大量应用请求时，会由基础设施管理平台识别并通过自动提供所需要的无状态、容器化计算环境，并在运行完成后自动释放。

从应用模式角度看，基于无服务器架构的应用系统通常采用基于消息机制的事件触发策略，并通过隐式调用模式完成事件响应。

三、考生需结合自身参与软件开发项目的实际状况，描述该软件的架构，并明确说明软件架构为什么属于无服务器架构，具有无服务器架构的哪些特征。并结合项目开发实际，说明采用无服务器架构模式后对软件开发过程的影响以及遇到的问题，包括业务代码开发、业务功能部署、系统水平扩展、业务交互方式等。

试题四 论软件质量保证及其应用

软件质量保证(Software Quality Assurance，SQA)是指为保证软件系统或软件产品充分满足用户要求的质量而进行的有计划、有组织的活动，这些活动贯穿于软件生产的整个生命周期。质量保证人员负责质量保证的计划、监督、记录、分析及报告工作，辅助软件开发人员得到高质量的最终产品。

请围绕“软件质量保证及其应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与管理和开发的软件项目以及你在其中所担任的主要工作。

2. 详细论述软件质量保证中常见的活动有哪些?阐述每个活动的主要内容。

3. 结合你具体参与管理和开发的实际项目，说明是如何实施软件质量保证的各项活动，说明其实施过程及应用效果。

试题四 写作要点

一、简要描述所参与管理和开发的软件系统开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、详细论述软件质量保证中常见的活动有哪些?阐述每个活动的主要内容。

软件质量保证活动包含有计划、监督、记录、分析及报告的软件质量保证活动，这些活动往往由一个独立的 SQA 小组执行。

（1）制订 SQA 计划

SQA 计划在制定项目计划时制定，它规定了软件开发小组和质量保证小组需要执行的质量保证活动。

（2）参与开发该软件项目的软件过程描述

软件开发小组为将要开展的工作选择软件过程，SQA 小组则要评审过程说明，以保证该过程与企业政策、内部的软件标准、外界所制定的标准以及项目开发计划的其他部分相符。

（3）评审

评审各项软件工程活动，核实其是否符合已定义的软件过程。SQA 小组识别、记录和跟踪所有偏离过程的偏差，核实其是否已经改正。

（4）审计

审计指定的软件工作产品，核实其是否符合已定义的软件过程的相应部分。SQA 小组对选出的产品进行评审，识别、记录和跟踪出现的偏差，核实其是否已经改正，定期向项目负责人报告结果。

（5）记录并处理偏差

确保软件工作及工作产品中的偏差已被记录在案，并根据预定规程进行处理。偏差可能出现在

项目计划、过程描述、采用的标准或技术工作产品中。

（6）报告

记录所有不符部分，并向上级管理部门报告。跟踪不符合的部分直到问题得到解决。除了进行上述活动外，SQA 小组还需要协调变更的控制与管理，并帮助收集和分析软件度量的信息。

三、结合你具体参与管理和开发的实际项目，说明是如何实施软件质量保证的各项活动，说明其实施过程及应用效果。

4.9 2016 下半年

试题一 论软件系统架构评估

对于软件系统，尤其是大规模的复杂软件系统来说，软件的系统架构对于确保最终系统的质量具有十分重要的意义，不恰当的系统架构将给项目开发带来高昂的代价和难以避免的灾难。对一个系统架构进行评估，是为了：分析现有架构存在的潜在风险，检验设计中提出的质量需求，在系统被构建之前分析现有系统架构对于系统质量的影响，提出系统架构的改进方案。架构评估是软件开发过程中的重要环节。

请围绕“论软件系统架构评估”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与架构评估的软件系统，以及在评估过程中所担任的主要工作。

2.分析软件系统架构评估中所普遍关注的质量属性有哪些?详细阐述每种质量属性的具体含义。

3.详细说明你所参与的软件系统架构评估中，采用了哪种评估方法，具体实施过程和效果如何。

试题一 写作要点

一、简要描述所参与架构评估的软件系统，并明确指出在评估过程中承担的主要工作。

二、分析软件系统架构评估中所普遍关注的质量属性，并详细阐述每种质量属性的具体含义。

系统架构评估中普遍关注的质量属性包括：

（1）性能

性能是指系统的响应能力，即需要多长时间才能对某个事件做出响应，或者在某段事件内系统所能处理的事件个数。经常用单位事件内所处理事务的数量或系统完成某个事务处理所需的时间来对性能进行定量表示。

（2）可靠性

可靠性是软件系统在应用或者系统错误面前，在意外或者错误使用的情况下维持软件系统的功能特性的基本能力。

（3）可用性

可用性是系统能够正常运行的时间比例。经常用两次故障之间的时间长度或在出现故障时系统能够恢复正常的速度来表示。

（4）安全性

安全性是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。

（5）可修改性

可修改性是指能够快速地以较高的性能价格比对系统进行变更的能力，包括可维护性、可扩展性、结构重构、可移植性。

（6）功能性

功能性是系统所能完成所期望的工作的能力。一项任务的完成需要系统中许多或大多数构件的相互协作。

（7）可变性

可变性是指体系结构经扩充或变更而成为新体系结构的能力。

（8）互操作性

互操作性是指作为系统组成部分的软件不是独立存在的，经常与其他系统或自身环境相互作用。如程序和用其他编程语言编写的软件系统的交互作用就是互操作性的问题。

三、针对作者实际参与的软件系统架构评估，说明所采用的评估方法，并描述其具体实施过程和效果。

现软件评估中的主要评估方法包括 SAAM(Scenarios-based Architecture Analysis Method)和 ATAM(Architecture Trade off Analysis Method，体系结构权衡分析方法)。作者可选择某种评估方法展开实际项目的系统评估。

试题二 论软件设计模式及其应用

软件设计模式(Software Design Pattern)是一套被反复使用的、多数人知晓的、经过分类编目的代码设计经验的总结。使用设计模式是为了重用代码以提高编码效率、增加代码的可理解性、保证代码的可靠性。软件设计模式是软件开发中的最佳实践之一，它经常被软件开发人员在面向对象软件开发过程中所采用。项目中合理地运用设计模式可以完美地解决很多问题，每种模式在实际应用中都有相应的原型与之相对，每种模式都描述了一个在软件开发中不断重复发生的问题，以及对应该原型问题的核心解决方案。

请围绕“论软件设计模式及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与分析和开发的软件系统，以及你在项目中所担任的主要工作。

2.说明常用的软件设计模式有哪几类?阐述每种类型特点及其所包含的设计模式。

3.详细说明你所参与的软件系统开发项目中，采用了哪些软件设计模式，具体实施效果如何。

试题二 写作要点

一、简要描述所参与分析和开发的软件系统开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、说明软件系统设计中常用的软件设计模式有哪几类，阐述每种类型的特点及其所包含的设计模式。

常用的软件设计模式主要包括：

（1）创建型模式

该类模式是对对象实例化过程的抽象，它通过采用抽象类所定义的接口，封装了系统中对象如何创建、组合等信息。

所包括的模式：Abstract Factory(抽象工厂)、Builder(建造者)、Factory Method(工厂方法)、Prototype(原型)、Singleton(单例)。

（2）结构型模式

该类模式主要用于如何组合已有的类和对象以获得更大的结构，一般借鉴封装、代理、继承等概念将一个或多个类或对象进行组合、封装，以提供统一的外部视图或新的功能。

所包括的模式：Adapter(适配器)、Bridge(桥接)、Composite(组合)、Decorator(装饰)、Facade(外观)、Flyweight(享元)、Proxy(代理)。

（3）行为型模式

该类模式主要用于对象之间的职责及其提供的服务的分配，它不仅描述对象或类的模式，还描述它们之间的通信模式，特别是描述一组对等的对象怎样相互协作以完成其中任一对象都无法单独完成的任务。

所包括的模式：Chain of Responsibility(职责链)、Command(命令)、Interpreter(解释器)、Iterator(迭代器)、Mediator(中介者)、Memento(备忘录)、Observer(观察者)、State(状态)、Strategy(策略)、Template Method(模板方法)、Visitor(访问者).

三、针对作者实际参与的软件系统开发项目，说明所采用的软件设计模式，并描述这些设计模式所产生的实际应用效果。

使用设计模式的作用主要表现在：(1)简化并加快设计；(2)方便开发人员之间的通信；(3)降低风

险；(4)有助于转到面向对象技术。

试题三 论数据访问层设计技术及其应用

在信息系统的开发与建设中，分层设计是一种常见的架构设计方法，区分层次的目的是为了实现“高内聚低耦合”的思想。分层设计能有效简化系统复杂性，使设计结松清晰，便于提高复用能力和产品维护能力。一种常见的层次划分模型是将信息系统分为表现层、业务逻辑层和数据访问层。信息系统一般以数据为中心，数据访问层的设计是系统设计中的重要内容。数据访问层需要针对需求，提供对数据源读写的访问接口；在保障性能的前提下，数据访问层应具有良好的封装性、可移植性，以及数据库无关性。

请围绕“论数据访问层设计技术及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的与数据访问层设计有关的软件项目，以及你在其中所担任的主要工作。

2.详细论述常见的数据访问层设计技术及其所包含的主要内容。

3.结合你参与管理和开发的实际项目，具体说明采用了哪种数据访问层设计技术，并叙述具体实施过程以及应用效果。

试题三 写作要点

一、简要叙述所参与管理和开发的软件项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、常见的数据访问层设计技术有 5 种数据访问模式。

（1）在线访问：该模式是基本的数据访问模式，在软件系统中不存在专门的数据访问层，由业务程序直接读取数据，与后台数据源进行交互。

（2）Data Access Object：DAO 模式是标准 J2EE 设计模式之一，该模式将底层数据访问操作与高层业务逻辑分离开。具体的 DAO 类包含访问特定数据源数据的逻辑。

（3）Data Transfer Object：DTO 是经典 EJB 设计模式之一。DTO 本身是一组对象或是数据的容器，它需要跨越不同进程或者网络的边界来传输数据。这类对象通常本身不包括具体的业务逻辑，对象内部仅进行一些诸如内部一致性检查和基本验证之类的方法。

（4）离线数据模型：是以数据为中心，数据从数据源获取后，将按照某种预定义的结构(如 IBMSDO 的 Data 图表结构或 ADO.NET 中的关系结构)存放在系统中，成为应用的中心。其特点是：①离线，数据操作独立于后台数据源；②与 XML 集成，数据可以方便地与 XML 格式文档相互转换。

（5）对象/关系映射(Object/Relation Mapping)：ORM 是一种工具、中间件或平台，它能够帮助

将应用程序中的数据转换成关系数据库中的记录；或者是将关系数据库中的记录转换成应用程序中代码便于操作的对象，使得程序员在开发过程中仅仅面对一个对象的概念，降低了对程序员数据库知识的要求，简化了数据库相关的开发工作。

三、考生需结合自身参与项目的实际状况，指出其参与管理和开发的项目中所进行的具体的数据访问层设计，说明具体的设计过程、使用的方法和工具，并对实际应用效果进行分析。

试题四 论微服务架构及其应用

近年来，随着互联网行业的迅猛发展，公司或组织业务的不断扩张，需求的快速变化以及用户量的不断增加，传统的单块(Monolithic)软件架构面临着越来越多的挑战，已逐渐无法适应互联网时代对软件的要求。在这一背景下，微服务架构模式(Micro service Architecture Pattern)逐渐流行，它强调将单一业务功能开发成微服务的形式，每个微服务运行在一个进程中，采用 HTTP 等通用协议和轻量级 API 实现微服务之间的协作与通信。这些微服务可以使用不同的开发语言以及不同数据存储技术，能够通过自动化部署工具独立发布，并保持最低限制的集中式管理。

请围绕“论微服务架构及其应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与管理和开发的、采用微服务架构的软件开发项目及在其中所担任的主要工作。

2. 与单块架构相比较，微服务架构有哪些特点?请列举至少 4 个特点并进行说明。

3. 结合你参与管理和开发的软件开发项目，描述该软件的架构，说明该架构是如何采用微服务架构模式的，并说明在采用微服务架构后，在软件开发过程中遇到的实际问题和解决方案。

试题四 写作要点

一、叙述你参与管理和开发的、采用微服务架构的软件开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、与单块架构相比，微服务架构具有如下特点：

（1）通过服务实现组件化。单个微服务实现简单，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。微服务能够被一个开发人员理解、修改和维护，这样小才队能够更关注自己的工作成果，并降低沟通成本。

（3）围绕业务功能构建开发团队。采用微服务架构，可以围绕业务功能构建开发团队，这样更符合企业的分工与组织结构，便于管理。

（4）支持多种开发语言与多种平台。不同的微服务能使用不同的语言开发，运行在不同的操作系统平台上，通过标准的协议和数据格式进行交互与协作。

（5）离散化数据管理。在微服务架构中，无法创建或维护统一的数据模型或结构，全局数据模

型将在不同的系统之间有所区别，需要进行数据模型的离散化管理。

（6）基础设施自动化。微服务强调以灵活的方式集成自动部署，通过持续集成工具实现基础设施自动化。

三、考生需结合自身参与软件开发项目的实际状况，描述该软件的架构，并明确说明软件架构为什么属于微服务架构，具有微服务架构的哪些特征。并结合项目开发实际，说明采用微服务架构模式后对软件开发过程的影响以及遇到的问题，包括服务的定义与划分、服务之间的协作关系、服务部署、服务管理等。