

2018 年系统架构师考试科目三：论文

1. 论软件开发过程 RUP 及其应用

RUP (Rational Unified Process) 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。

【解析】

一、应结合自己参与的信息系统项目，说明在其中所承担的工作。

二、本文内容的组织可以将问题 2 与问题 3 结合起来论述。先说明 RUP 的四个阶段及 RUP 的特征，然后再论述每个阶段，作者开展了哪些工作。

RUP 中的软件过程在时间上被分解为四个顺序的阶段，分别是初始阶段、细化阶段、构建阶段和交付阶段。每个阶段结束时都要安排一次技术评审，以确定这个阶段的目标是否已经满足。如果评审结果令人满意，就可以允许项目进入下一个阶段。

四个阶段的核心任务分别为：

(1) 初始阶段

明确地说明项目规模。这涉及了解环境及最重要的需求和约束，以便于可以得出最终产品的验收标准。

计划和准备商业理由。评估风险管理、人员配备、项目计划和成本/进度/收益率折中的备选方案。

综合考虑备选构架，评估设计和自制/外购/复用方面的折中，从而估算出成本、进度和资源。此处的目标在于通过对一些概念的证实来证明可行性。该证明可采用可模拟需求的模型形式或用于探索被认为高风险区域的初始原型。初始阶段的原型设计工作应该限制在确信解决方案可行就可以了。该解决方案在细化和构建阶段实现。

准备项目的环境，评估项目和组织，选择工具，决定流程中要改进的部分。

(2) 细化阶段

快速确定构架，确认构架并为构架建立基线。

根据此阶段获得的新信息改进前景，对推动构架和计划决策的最关键用例建立可靠的了解。

为构建阶段创建详细的迭代计划并为其建立基线。

改进开发案例，定位开发环境，包括流程和支持构建团队所需的工具和自动化支持。

改进构架并选择构件。评估潜在构件，充分了解自制/外购/复用决策，以便有把握地确定构建阶段的成本和进度。集成了所选构架构件，并按主要场景进行了评估。通过这些活动得到的经验有可能导致重新设计构架、考虑替代设计或重新考虑需求。

(3) 构建阶段

资源管理、控制和流程优化。

完成构件开发并根据已定义的评估标准进行测试。

根据前景的验收标准对产品发布版进行评估。

(4) 产品化阶段（提交阶段）

执行部署计划。

对最终用户支持材料定稿。

在开发现场测试可交付产品。

制作产品发布版。

获得用户反馈。

基于反馈调整产品。

使最终用户可以使用产品。

RUP 最核心的 3 个特征是：用例驱动、以架构为中心的、迭代和增量。

制品 (Artifact) —— what 的问题：制品是活动生成、创建或修改的一段信息。也可译为产品、工件等，和制品的意思差不多。

workflow (Workflow) —— when 的问题： workflow 描述了一个有意义的连续的活动序列，每个 workflow 产生一些有价值的产品，并显示了角色之间的关系。

2. 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。

【解析】

一、应结合自己参与的信息系统项目，说明在其中所承担的工作。

二、首先需要弄清楚的是此处的“软件体系结构演化”实际上指的是 ABSD 方法中的最后一个阶段。体系结构演化是使用系统演化步骤去修改应用，以满足新的需求。主要包括以下六个步骤。

1、需求变动归类

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，以对应这部分变化的需求。

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 1 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、更新构件的相互作用

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，则需要第 2 到第 6 步之间进行迭代。

三、论文中需要结合项目实际工作, 阐述 6 个步骤的具体应用, 此时可以重点讲述其中的 2-3 个方面, 不必面面俱到的论述, 最后说明实施效果。

3. 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。通过实施基于 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。

【解析】

一、应结合自己参与的信息系统项目, 说明在其中所承担的工作。

二、面向服务架构的主要技术有 Web 服务、ESB。涉及到的标准有:

1、UDDI 协议

UDDI (统一描述、发现和集成协议) 计划是一个广泛的、开放的行业计划, 它使得商业实体能够 (1) 彼此发现, (2) 定义他们怎样在 Internet 上互相作用, 并在一个全球的注册体系架构中共享信息。UDDI 是这样一种基础的系统构筑模块, 他使商业实体能够快速、方便地使用他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 同时也是 Web 服务集成的一个体系框架。它包含了服务描述与发现的标准规范。UDDI 规范利用了 W3C 和 Internet 工程任务组织 (IETF) 的很多标准作为其实现基础, 比如扩展标注语言 (XML)、HTTP 和域名服务 (DNS) 等协议。另外, 在跨平台的设计特性中, UDDI 主要采用了已经被提议给 W3C 的 SOAP (Simple Object Access Protocol, 简单对象访问协议) 规范的早期版本。

2、WSDL 规范

WSDL 是 Web Services Description Language (Web 服务描述语言) 的缩写, 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, 由 Ariba、Intel、IBM、MS 等共同提出, 通过 WSDL, 可描述 Web 服务的三个基本属性:

- 1、服务做些什么——服务所提供的操作 (方法);
- 2、如何访问服务——和服务交互的数据格式以及必要协议;
- 3、服务位于何处——协议相关的地址, 如 URL。

WSDL 文档以端口集合的形式来描述 Web 服务, WSDL 服务描述包含对一组操作和消息的一个抽象定义, 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。

3、SOAP 协议

SOAP (Simple Object Access Protocol) 简单对象访问协议是在分散或分布式的环境中交换信息的简单的协议, 是一个基于 XML 的协议。它包括四个部分: SOAP 封装 (Envelope), 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 (Encoding Rules), 用于表示应用程序需要使用的数据类型的实例; SOAP RPC 表示 (RPC Representation), 表示远程过程调用和应答的协定; SOAP 绑定 (Binding), 使用底层协议交换信息。

三、论文中需要结合项目实际工作, 论述构建 SOA 架构时遇到的问题以及如何解决的, 效果如何。注意本主题才是文章的重心所在。

4. 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。

【解析】

一、应结合自己参与的信息系统项目, 说明在其中所承担的工作。

二、

NoSQL 的主要优势:

- (1) 避免不必要的复杂性
- (2) 高吞吐量
- (3) 高水平扩展能力和低端硬件集群
- (4) 避免了昂贵的对象-关系映射

NoSQL 的缺点:

- (1) 数据模型和查询语言没有经过数学验证
- (2) 不支持 ACID 特性
- (3) 功能简单
- (4) 没有统一的查询模型

NoSQL 数据库的四大分类:

1、键值(Key-Value)存储数据库

这一类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。例如: Tokyo Cabinet/Tyrant, Redis, Voldemort, Oracle BDB。

2、列存储数据库。

这部分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。

HBase: HBase 是一个分布式的、面向列的开源数据库, 该技术来源于 Fay Chang 所撰写的 Google 论文“Bigtable: 一个结构化数据的分布式存储系统”。就像 Bigtable 利用了 Google 文件系统 (File System) 所提供的分布式数据存储一样, HBase 在 Hadoop 之上提供了类似于 Bigtable 的能力。HBase 是 Apache 的 Hadoop 项目的子项目。HBase 不同于一般的关系数据库, 它是一个适合于非结构化数据存储的数据库。另一个不同的是 HBase 基于列的而不是基于行的模式。

3、文档型数据库

文档型数据库的灵感是来自于 Lotus Notes 办公软件的,而且它同第一种键值存储相类似。该类型的数据库模型是版本化的文档,半结构化的文档以特定的格式存储,比如 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。国内也有文档型数据库 SequoiaDB,已经开源。

Mongo DB: Mongo DB 是目前在 IT 行业非常流行的一种非关系型数据库(NoSql),其灵活的数据存储方式备受当前 IT 从业人员的青睐。Mongo DB 很好的实现了面向对象的思想(OO 思想),在 Mongo DB 中 每一条记录都是一个 Document 对象。Mongo DB 最大的优势在于所有的数据持久操作都无需开发人员手动编写 SQL 语句,直接调用方法就可以轻松的实现 CRUD 操作。

Sequoia DB: SequoiaDB 是一款分布式非关系型文档数据库,可以被用来存取海量非关系型的数据,其底层主要基于分布式,高可用,高性能与动态数据类型设计 SequoiaDB 可以独立作为一款高性能可扩展的 NoSQL 数据库使用,也可与当前主流分布式计算框架 Hadoop 紧密集成。

4. 图形(Graph)数据库

图形结构的数据库同其他行列以及刚性结构的 SQL 数据库不同,它是使用灵活的图形模型,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制定数据模型。许多 NoSQL 数据库都有 REST 式的数据接口或者查询 API。如: Neo4J, InfoGrid, Infinite Graph。

三、论文中需要结合项目实际工作,详细论述在项目中所采用的 noSQL 数据库,并详细说明实施效果。