

银行业务介绍 常见应用系统介绍 --3

柳金峰
2011.11

- **第三部分金融软件开发模型与过程**
- 软件开发模型
- 软件开发过程

● 软件开发模型

软件开发模型是指软件开发全部过程、活动和任务的结构框架，目前软件开发实践中使用的各种开发模型，都是下面这些基本组成部分的不同的排列与组合。

- 市场分析，可行性研究，与项目定义
- 需求分析
- 设计（概要设计和详细设计）
- 编码实现
- 测试
- 使用与维护

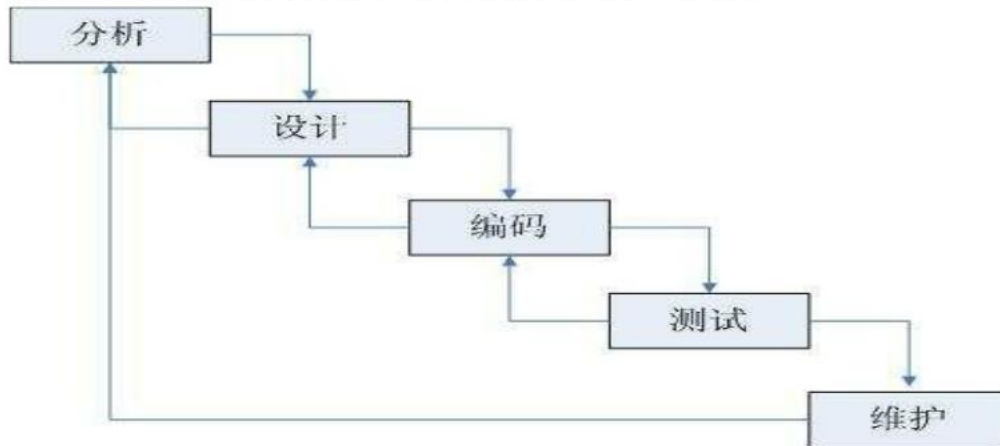
● 软件开发模型

➤ 主要有以下几种模型

- **瀑布模型**，强调开发的阶段性，同时强调早期计划及需求调查的重要性。这种模型一般用于做一些需求已明确的并很少变化的软件
- **演化模型**，该模型主要针对事先不能完整定义需求的软件开发。用户可以给出待开发系统的核心需求，并且当看到核心需求实现后，能够有效地提出反馈，以支持系统的最终设计和实现。软件开发人员根据用户的反馈，实施开发的迭代过程。第一迭代过程均由需求、设计、编码、测试、集成等阶段组成，为整个系统增加一个可定义的、可管理的子集。采用这种开发模型，开发中的经验教训能反馈应用于本产品的下一个循环过程，大大提高质量与效率
- **螺旋模型**，是瀑布模型与演化模型相结合，并加入两者所忽略的风险分析所建立的一种软件开发模型。螺旋模型基本的做法是在“瀑布模型”的每一个开发阶段之前，引入非常严格的风险识别、风险分析和风险控制。直到采取了消除风险的措施之后，才开始计划下一阶段的开发工作。否则，项目就很可能被取消

● 软件开发过程

- 瀑布型软件开发过程可以包括以下五个阶段：



- 在实际开发过程中，软件开发并不是从第一步进行到最后一步，而是在任何阶段，在进入下一阶段前一般都有一步或几步的回溯。如在测试过程中的问题可能要求修改设计，用户可能会提出一些需要来修改需求说明书等

● 软件开发过程一分析

- 软件需求分析就是回答做什么的问题。它是一个对用户的需求进行去粗取精、去伪存真、正确理解，然后把它用软件开发语言（即需求规格说明书）表达出来的过程
- 金融软件的需求管理是一个综合性的过程，应做到以下几点：
 - 负责制：应用部门和开发部门都应实行，两个部门都要有专门的机构负责从本部门的角度进行需求管理，由专人负责，有专门的部门领导负责协调，并对需求中出现的各种问题和错误负责。需求管理涉及后续各个方面，直接关系到软件产品的最终质量，因此必须强化需求负责制，确保需求及其变更始终处于良好的管理之下。
 - 规范化：金融软件需求管理应遵循科学规范的原则，可以避免非程序性、随意性等多方面问题。在需求管理中，对需求的获取、需求分析、需求分析的描述（《软件需求规格说明书》及其它文档）、需求的变更等需求管理的各方面制定相应的管理规范，并在工作中加以完善，坚持执行
 - 严肃性与灵活性：业务需求的提出及变更是一件严肃的事情。需求管理的目标之一，就是减少需求的变动，维护需求的相对稳定性。需求的每一处变动，都会对后续的开发工作产生影响，甚至导致某些工作推倒重来。因此必须维护需求的严肃性，不允许随意变更需求的内容。如确有必要，应经过变更需求的管理程序。对于业务上某些不影响原则问题的细节调整，开发部门可以根据开发工作的实际情况，在符合需求的大框架内予以满足，并将变更的内容及时归档记录，作为《软件需求规格说明书》的附件，从而在需求管理上体现出一定的灵活性

● 软件开发过程一分析

- 软件需求直接关系到软件产品的质量。一个好的软件需求应具有如下特性：
 - **完整性**：要从全局出发，不能单纯从本业务考虑问题。一方面要完整地反映该项业务，另一方面还要全面反映本项业务同其它关联业务的联系。
 - **准确性**：准确无误，无二义，各项要求、业务做法、每种处理的详细流程、数据等方面的要求等明确定义，不能模棱两可、含糊不清。
 - **通用性**：业务需求要具有较广泛的适应性，要能够适应大部分分支机构、适应大部分业务处理情况，减少以后各分支机构对系统的修改要求。
 - **前瞻性**：业务需求要具有前瞻性，要能够反映该项业务当前的发展状况（包括同业情况）和发展趋势。系统要留有可扩充的余地。
 - **稳定性**：一定时限内相对稳定、不变。
 - **权威性**：业务需求要具有权威性，能被普遍接受，并具有很强的约束力。
 - **可行性**：需求在技术实现和经济负担上要符合实际，切实可行。
 - **安全性**：金融业在社会经济生活中的特殊性对金融软件的安全性提出了较高的要求，从需求的提出就应充分考虑软件的安全性问题，要有专门负责安全生产或稽核的人员全程参与需求管理及软件开发

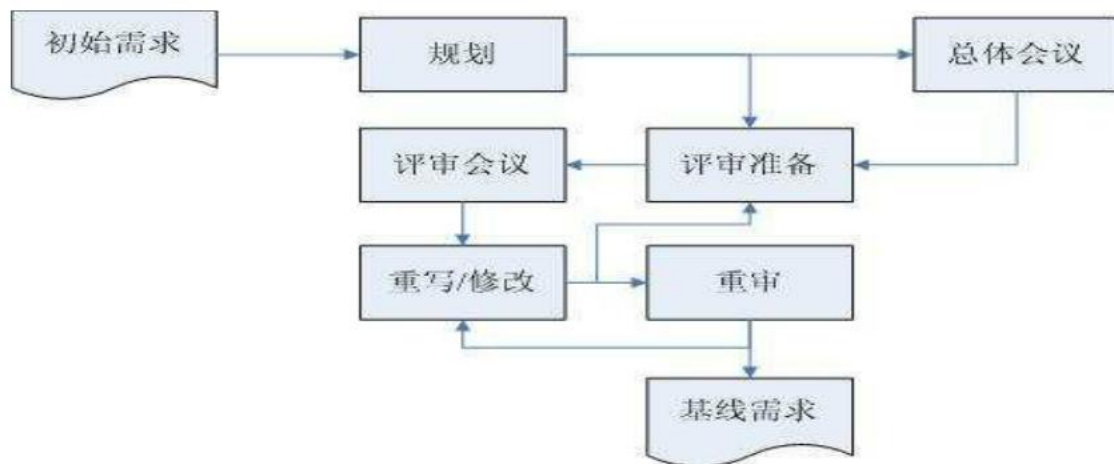
● 软件开发过程—分析

➤ 通常可以将需求分为三类：

- 期望需求：指那些客户或许并未提及，但若缺少，会让他们感到不满意的需求。
- 普通需求
- 兴奋需求，指那些实现了会给客户带来惊喜，但若未实现也不会受到责备的需求。

● 软件开发过程一分析

- 需求分析完成后，不能直接进入分析阶段，还需要进行验证和审核的动作，验证的目的有如下几点：
 - 需求规格说明正确描述了预期的系统行为和特征。
 - 需求是完整和准确的。
 - 对需求的所有看法都是一致的，不矛盾的。
- 需求可以为继续进行设计，编码和测试提供足够的支持。
- 需求评审过程如下图：



● 软件开发过程—设计

- 进入软件设计阶段前，首先要根据系统的性能要求，预算，开发周期，安全性等综合因素进行软件系统架构和数据库平台以及开发语言（工具）的选型
- 在选择数据库平台时要考虑以下几个方面：
 - 具有很高的稳定性，可靠性和可伸缩性。数据库不仅应适用于各种硬件平台，而且在大数据量的条件下，能保持数据处理的高效性和数据的可靠性。
 - 基于开放式的体系结构，为保障业务系统的发展和扩充数据库管理系统，其体系结构是一个多平台，开放式系统，能够支持多种硬件平台和操作系统，支持多种网络通信协议，能方便地同其他异种数据交换所需要的信息，硬件平台升级后，软件能方便地移植到新的硬件平台上。
 - 统一的，易操作的数据库管理平台，数据库管理系统应是基于图形界面的管理平台，易于管理和维护，方便系统的安装，升级和版本控制，将大部分的数据库日常维护操作，如数据备份，恢复，故障检测等工作，交由系统自动完成，减少人为操作成本和不安全性。
 - 具有可靠的数据库安全性控制机制，保证数据库访问权限的有效控制，防止未授权实体访问任何数据，保证数据库中数据的完整性和一致性。防止多个授权用户同时访问相同数据时产生冲突。系统同时具有容错性，以便使设备出现故障时也不会影响系统运行，一旦出现灾难，具有具备完善的灾难恢复能力。
 - 数据库应支持超大数据量的存储和管理，对大型商业银行来说，数据应具备TB或PB级的数据存储能力

● 软件开发过程—设计

- 进入软件设计阶段前，首先要根据系统的性能要求，预算，开发周期，安全性等综合因素进行软件系统架构和数据库平台以及开发语言（工具）的选型
- 在选择数据库平台时要考虑以下几个方面：
 - 具有很高的稳定性，可靠性和可伸缩性。数据库不仅应适用于各种硬件平台，而且在大数据量的条件下，能保持数据处理的高效性和数据的可靠性。
 - 基于开放式的体系结构，为保障业务系统的发展和扩充数据库管理系统，其体系结构是一个多平台，开放式系统，能够支持多种硬件平台和操作系统，支持多种网络通信协议，能方便地同其他异种数据交换所需要的信息，硬件平台升级后，软件能方便地移植到新的硬件平台上。
 - 统一的，易操作的数据库管理平台，数据库管理系统应是基于图形界面的管理平台，易于管理和维护，方便系统的安装，升级和版本控制，将大部分的数据库日常维护操作，如数据备份，恢复，故障检测等工作，交由系统自动完成，减少人为操作成本和不安全性。
 - 具有可靠的数据库安全性控制机制，保证数据库访问权限的有效控制，防止未授权实体访问任何数据，保证数据库中数据的完整性和一致性。防止多个授权用户同时访问相同数据时产生冲突。系统同时具有容错性，以便使设备出现故障时也不会影响系统运行，一旦出现灾难，具有具备完善的灾难恢复能力。
 - 数据库应支持超大数据量的存储和管理，对大型商业银行来说，数据应具备TB或PB级的数据存储能力
- 数据库设计包括表结构设计，表空间设计，索引设计，存储策略设计，数据库操作权限设计等等

● 软件开发过程—设计

- 软件设计的主要任务就是将软件分解成能实现某个功能的数据和程序说明、可执行程序的程序单元，然后进行模块设计。
- 软件设计可以分为概要设计和详细设计两个阶段：
 - 概要设计就是结构设计，其主要目标就是给出软件的模块结构，用软件结构图表示。
 - 详细设计的首要任务就是设计模块的程序流程、算法和数据结构，数据库的设计也可以归为软件设计的一部分，内容包括数据库选型，表结构设计，索引设计，表空间设计等。
 - 银行业使用的操作系统一般是UNIX（如HP-UX，AIX，OS400，OS390等），Linux, Windows，数据库往往使用ORACLE，DB2，INFORMIX，SYBASE，SQLSERVER等。应用平台包括AS400, RS6000, RS9000, HP9000等。
 - 应用复杂度、性能、安全性、成本等要求的不同，往往导致选择操作系统、中间件、数据库所考虑的出发点也不同

● 软件开发过程—编码

- 软件编码是指把软件设计转换成计算机可以接受的程序，即写成以某一程序设计语言表示的“源程序清单”。充分了解软件开发语言、工具的特性和编程风格，有助于开发工具的选择以及保证软件产品的开发质量。在银行业软件设计中，后台交易的处理用C，COBOL语言实现居多，而前台业务系统的设计往往使用J2EE，.NET，PB，C等语言，为了在数据库层面提高数据运算的速度，也有不少应用直接采用数据库存储过程。
- 在编码阶段，需要强调的就是编码的规范性，即开发人员个人的编码规范要与开发团队的编码规范相统一，对程序设计中需要特别说明的内容尽可能用注释或相关文档进行描述。

● 软件开发过程—测试

- 软件测试的目的是以较小的代价发现尽可能多的错误。来验证软件的正确性和可用性。
- 软件测试可以分为三个层次：功能验证，系统测试，性能测试
 - **功能验证**的目的在于通过测试验证软件的功能设计达到了软件需求中的功能性要求，要实现测试目标的关键在于设计一套出色的测试用例（测试数据和预期的输出结果组成了测试用例）。不同的测试方法有不同的测试用例设计方法。两种常用的测试方法是白盒法和黑盒法，白盒法测试对象是源程序，依据的是程序内部的逻辑结构来发现软件的编程错误、结构错误和数据错误。用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果。黑盒法依据的是软件的功能或软件行为描述，发现软件的接口、功能和结构错误。其中接口错误包括内部/外部接口、资源管理、集成化以及系统错误。黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。
 - **系统测试**的目的是把整个系统作为一个测试对象，通过系统内多个模块的协作运行，发现模块接口之间，协同之间是否存在一些问题
 - **性能测试**（也叫压力测试），主要测试软件的性能，包括负载测试，强度测试，基准测试。负载测试目的在于通过应用很大的工作负载来使软件超负荷运转。检验系统的关键性能指标（如最大并发连接数，相应时间，IO等）在可承受的范围内所能接受的最大负载，以此来评估系统的性能特征。在这个阶段经常能够发现许多隐蔽的错误，而这些问题往往是很难修复的；强度测试时在系统资源特别低的情况下，验证软件系统的性能特征，此类测试的目的是找出因资源不足或资源争用而导致的错误；基准测试是与先用的系统进行比较，主要检验本系统是否与类似的产品具有竞争性的一种测试。

● 软件开发过程—测试

- 常见的性能测试工具有LoadRunner和JMeter，性能测试一般分以下几个步骤：
 - ①熟悉待测软件，如这个软件是干什么的，其功能和性能主要体现在那几方面，有什么特点等。
 - ②制定测试计划，测试计划就是定义一个测试项目的过程，以便准确地度量和控制测试，测试计划包括准备采用什么测试工具，根据现有条件准备搭建的模拟测试环境，测试完成的标准，测试方法和进度安排等
 - ③实施测试，按照测试计划，在各种条件下允许事先设计的测试脚本，记录被测试软件的性能参数，在一定范围内调整数据库的大小，并发访问的用户数，持续访问时间等测试条件以获得所需要的数据。
 - ④分析测试结果，测试会收集到大量的数据。根据这些数据可以帮助分析系统性能。对系统性能的描述可采用曲线图，条形图，报表等多种直观的表现方式。

● 软件开发过程—维护

- 维护是旨在已完成对软件的研制（分析、设计、编码和测试）工作并交付使用以后，对软件产品所进行的修改，完善的活动。即根据软件运行的情况，对软件进行适当修改，以适应新的要求，以及纠正运行中发现的错误：
- 通常有四类维护活动：
 - 改正性维护，也就是诊断和改正在使用过程中发现的软件错误；
 - 适应性维护，即修改软件以适应环境的变化；
 - 完善性维护，即根据用户的要求改进或扩充软件使它更完善；
 - 预防性维护，即修改软件为将来的维护活动预先做准备。
- 虽然没有把维护阶段进一步划分成更小的阶段，但是实际上每一项维护活动都应该经过提出维护要求（或报告问题），分析维护要求，提出维护要求，提出维护方案，审批维护方案，确定维护计划，修改软件设计，修改程序，测试程序，复查验收等一系列步骤，因此实质上是经历了一次压缩和简化了的软件定义和开发的全过程。