

第2章 基本理论

本章内容：

2. 1 软件工程过程

2. 2 软件生命周期

2. 3 软件生命周期模型

2. 4 软件开发方法

2. 5 软件工具与开发

2.1 软件工程过程

软件工程由几个重要过程组成这个框架，这些过程含有用来获取、供应、开发、操作和维护软件所用的基本的、一致的要求。该框架还用来控制和管理软件的过程。各种组织和开发机构可以根据具体情况选择和剪裁，可在机构的内部或外部实施。软件工程过程包括如下七个过程：

1. 开发过程

开发过程就是开发者和机构为了定义和开发软件或服务所需的活动。此过程包括需求分析、设计、编码、集成、测试、软件安装和验收等活动。

2. 管理过程

管理过程为软件工程过程中各项管理活动，包括项目开始和范围定义；项目管理计划；实施和控制；评审和评价；项目完成。

3. 供应过程

供应过程是供方按照合同向需求方提供合同中的系统、软件产品或服务所需的活动。

4. 获取过程

根据需要，获取过程是需求方按合同要求获取一个系统、软件产品或服务的活动。

5. 操作过程

操作过程是操作者和机构为了在规定的运行环境中为其用户运行一个计算机系统所需要的活动。

6. 维护过程

维护过程是维护者和机构为了管理软件的修改，使它处于良好运行状态需要的活动。

7. 支持过程

支持过程对项目的生命周期过程给予支持。它有助于项目的成功并能提高项目的质量。

2.2 软件生命周期

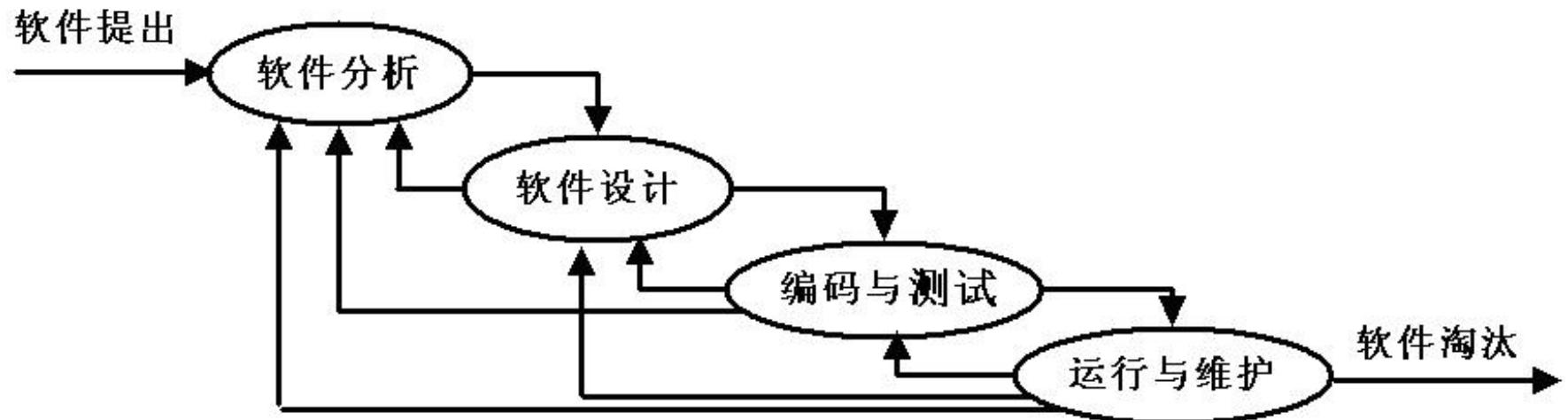
软件生命周期（Software Life Cycle），也称为软件生存周期，是软件工程最基础的概念。软件工程的方法、工具和管理都是以软件生命周期为基础的活动。换句话说，软件工程强调的是使用软件生命周期方法学和使用成熟的技术和方法来开发软件。

软件生命周期的基本思想是：任何一个软件都是从它的提出开始到最终被淘汰为止，有一个存在期。软件“生命周期”的概念并不是说软件同硬件一样，存在“被用坏”和“老化”问题，而是指其无存在价值。

软件生命周期一种典型的阶段划分为：问题定义、可行性研究、需求分析、概要设计（总体设计）、详细设计、编码、测试和维护等八个阶段。

软件生命周期内阶段的划分要受软件的规模、性质、种类、开发方法等影响，阶段划分过细还会增加阶段之间联系的复杂性和软件工作量，在实际软件工程项目较难操作。也有提出软件生命周期内划分成四个活动时期：软件分析时期、软件设计时期、编码与测试时期以及软件运行与维护时期。

软件生命周期内划分成四个活动时期：



2.2.1 软件分析时期

软件分析时期，也可称为软件定义与分析时期。这个时期的根本任务是确定软件项目的目标，软件应具备的功能和性能，构造软件的逻辑模型，并制定验收标准。

这个时期包括问题定义、可行性研究和需求分析三个阶段，可以根据软件系统的大小和类型决定是否细分阶段。

1. 可行性研究和项目开发计划

可行性研究和项目开发计划阶段必须要回答的问题是“要解决的问题是什么”。该问题有可行的解决办法吗？若有解决问题的办法，则需要多少费用？需要多少资源？需要多少时间？要回答这些问题，就要进行问题定义、可行性研究，制定项目开发计划。

2. 需求分析

需求分析阶段的任务不是具体地解决问题，而是准确地确定“软件系统必须做什么”，确定软件系统必须具备哪些功能。

2.2.2 软件设计时期

软件设计时期的根本任务是将分析时期得出的逻辑模型设计成具体计算机软件方案。具体来说，主要包括：

- (1) 设计软件的总体结构。
- (2) 设计软件具体模块的实现算法。
- (3) 软件设计结束之前，也要进行有关评审，评审通过后才能进入编码时期。

1. 概要设计

在概要设计阶段，开发人员要把确定的各项功能需求转换成需要的体系结构，在该体系结构中，每个成分都是意义明确的模块，即每个模块都和某些功能需求相对应。

这个阶段要考虑几种可能的方案：

- (1) 最低成本方案。系统完成最必要的工作。
- (2) 中等成本方案。不仅能够完成预定的任务，而且还有用户没有指定的功能。
- (3) 高成本方案。系统具有用户希望的所有功能的系统。

2. 详细设计

详细设计阶段就是为每个模块完成的功能进行具体描述，要把功能描述转变为精确的、结构化的过程描述。即该模块的控制结构是怎样的，先做什么，后做什么，有什么样的条件判定，有些什么重复处理等，并用相应的表示工具把这些控制结构表示出来。

2.2.3 编码与测试时期

编码与测试时期，也可称为软件实现时期。在这个时期里，主要是组织程序员将设计的软件“翻译”成计算机可以正确运行的程序；并且要经过按照软件分析中提出需求要求和验收标准进行严格的测试和审查。审查通过后才可以交付使用。

1. 编码

编码阶段就是把每个模块的控制结构转换成计算机可接受的程序代码，即写成以某特定程序设计语言表示的“源程序清单”。当然，写出的程序应是结构好，清晰易读，并且与设计相一致。

2. 测试

测试是保证软件质量的重要手段，其主要方式是在设计测试用例的基础上检验软件的各个组成部分。测试分为模块测试、组装测试、确认测试。

(1) 模块测试是查找各模块在功能和结构上存在的问题。

(2) 组装测试是将各模块按一定顺序组装起来进行的测试，主要是查找各模块之间接口上存在的问题。

(3) 确认测试是按软件需求说明书上的功能逐项进行的，发现不能满足用户需求的问题，决定开发的软件是否合格，能否交付用户使用等。用正式的文档将测试计划方案和实际结果保存下来作为软件配置的组成部分。

2.2.4 运行与维护时期

运行与维护时期，简称维护时期。维护是计算机软件不可忽视的重要特征。维护是软件生命周期中最长、工作量最大、费用最高的一项任务。事实上，软件工程的提出最主要的因素之一就是软件出现了难以维护这种“危机”。

软件活动时期划分主要有如下几个优点：

- (1) 每个软件活动时期的独立性较强，任务明确，且联系简单，容易分工。
- (2) 软件工程过程清晰，简明。
- (3) 软件规模大小都适合，大型软件可以在软件活动时期内在划分阶段进行。
- (4) 适合各种软件工程开发模型和开发方法。
- (5) 适合各类软件工程。

2.3 软件生命周期模型

软件生命周期模型是指开发软件项目的总体过程思路。

2.3.1 软件生命周期模型的概念

软件生命周期模型确立了软件开发和演绎中各阶段的次序限制以及各阶段活动的准则，确立开发过程应遵守的规定和限制，便于各种活动的协调以及各种人员的有效通信，有利于活动重用的活动管理。

目前有若干种软件生命周期模型，如瀑布模型、原型模型、增量模型、螺旋模型、喷泉模型、基于知识的模型和变换模型。

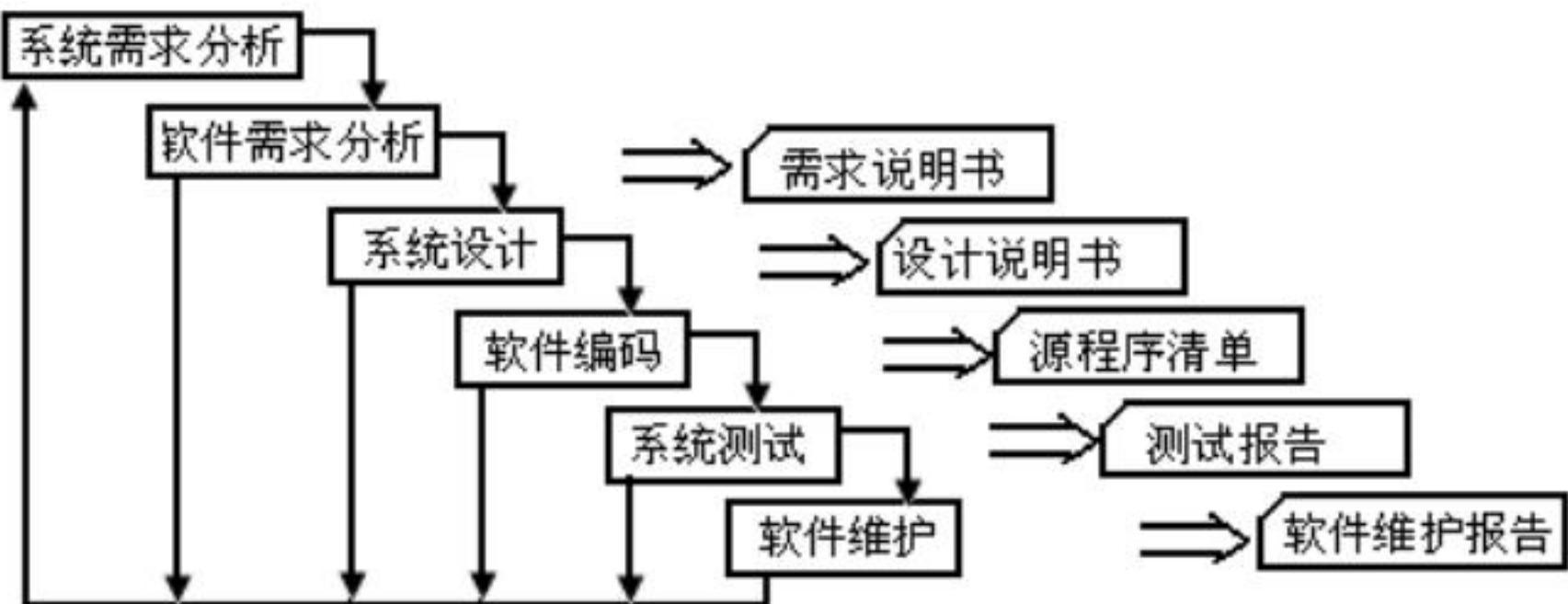
2.3.2 瀑布模型

瀑布模型是将软件生命周期各活动规定为依线性顺序联接的若干阶段的模型。

软件生命周期划分成七个阶段，包括：可行性分析、项目开发计划、需求分析、概要设计、详细设计、编码、测试和维护。每个阶段的工作完成都需要评审等技术确认。

瀑布模型为软件开发提供了一种有效的管理模型。根据这一模式制定开发计划，进行成本预算，组织开发力量，以项目的阶段评审和文档控制为手段有效地对整个开发过程进行指导。因此它是以文档作为驱动、适合于需求很明确的软件项目开发的模型。

1. 濑布模型



2. 瀑布模型的特点

瀑布模型是一种整体开发模型，在开发过程中，用户看不见系统是什么样，只有开发完成向用户提交整个系统时，用户就能看到一个完整的系统。

瀑布模型是以文档形式驱动的，为合同双方最终确认产品规定了蓝本，为管理者进行项目开发管理提供了基础，为开发过程施加了“政策”或纪律限制，约束开发过程中的活动。

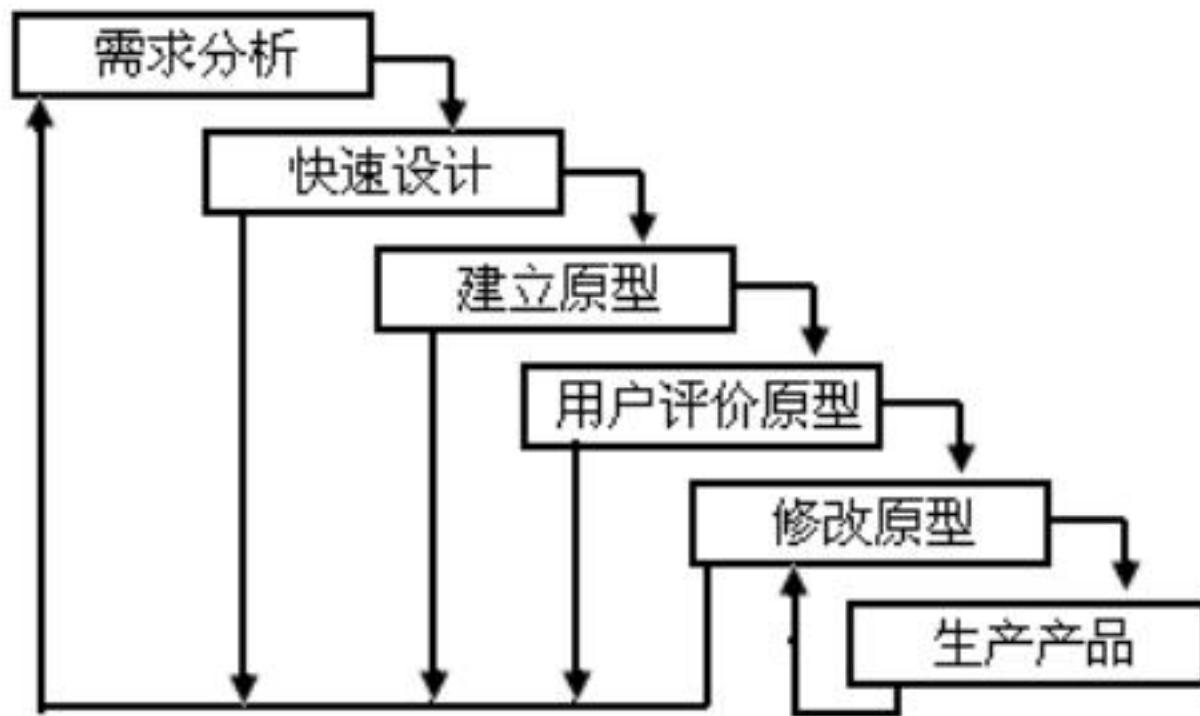
3. 瀑布模型的局限性

它是一种理想的线性开发模式，缺乏灵活性，特别是无法解决软件需求不明确或不准确的问题。这些缺点最终可能导致开发出的软件并不是用户真正需要的软件，并且问题往往是在开发过程完成后才能发现，已为时太晚。为此必须要进行返工，或者在运行中进行大量修改。这种返工或修改付出的代价巨大。同时，随着软件开发项目的规模日益扩大，瀑布模型缺乏灵活性的缺点引发的问题更为严重。

2.3.3 原型模型

原型模型（Prototyping Model）则是借助一些软件开发工具或环境尽可能快地构造一个实际系统的简化模型。

原型模型



原型模型的最大特点是：利用原型法技术能够快速实现系统的初步模型，供开发人员和用户进行交流，以便较准确获得用户的需求；采用逐步求精方法使原型逐步完善，这是一种在新的高层次上不断反复推进的过程，它可以大大避免在瀑布模型冗长的开发过程中看不见产品雏形的现象。

适合采用原型模型的条件：

- (1) 首先得有快速建立系统原型模型的软件工具与环境。
- (2) 原型模型适合于那些不能预先确切定义需求的软件开发。
- (3) 原型模型适合于那些项目组成员（包括分析员、设计员、程序员和用户等）不能很好协同配合、交流或通信上存在困难的情况。

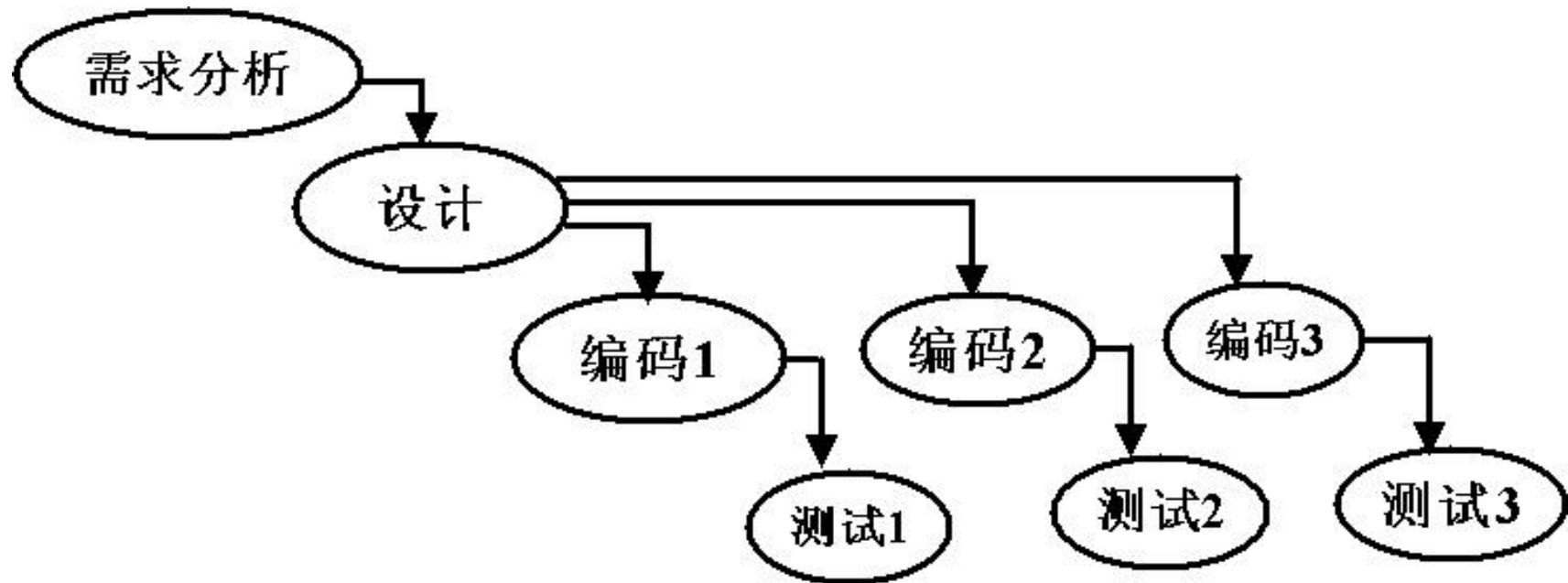
2.3.4 增量模型

增量模型是快速分析并构造一个小的原型系统，满足用户的某些要求后，使用户在使用过程中受其启发，逐步确定各种需求。

1. 增量构造模型

需求分析阶段和设计阶段按瀑布模型的整体方式开发。但是编码阶段和测试阶段按增量方式开发。这种模型的优点是开发中用户可以及早看到部分软件功能，发现问题，以便在开发其他软件功能时及时解决问题。

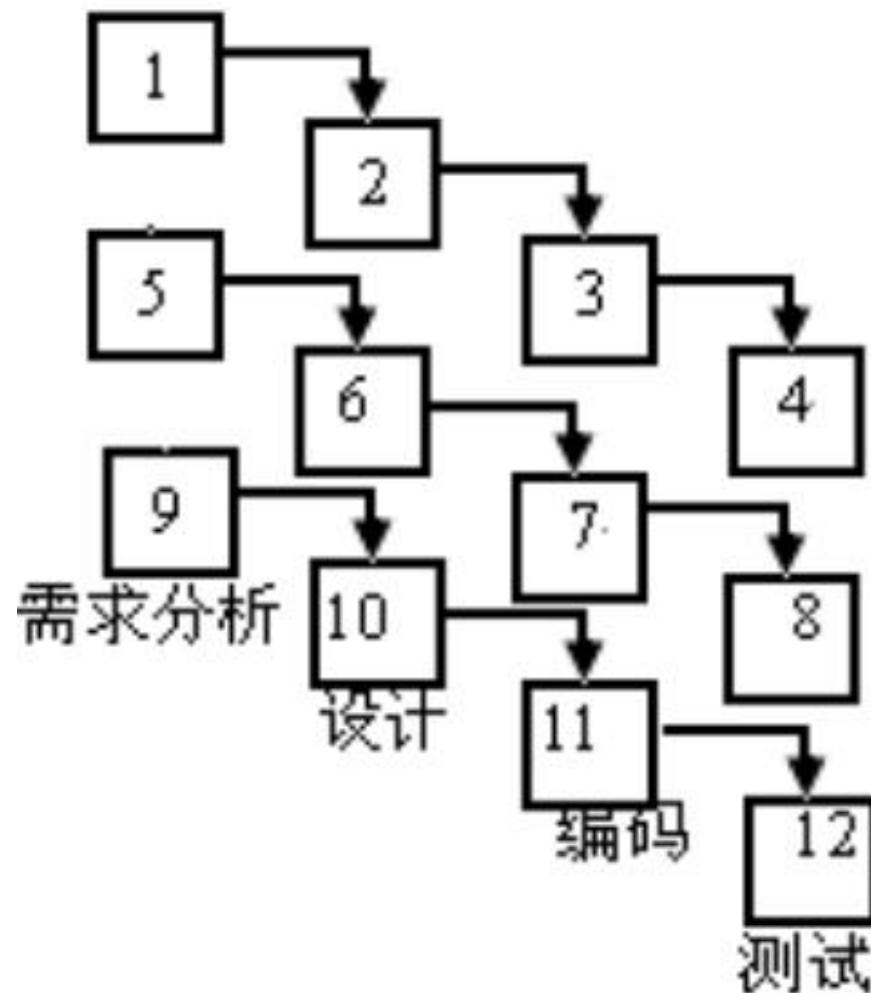
增量构造模型



2. 演化提交模型

先对某部分功能进行需求分析，然后按顺序进行设计、编码和测试，把该功能进行开发，提交用户直至所有功能全部增量开发完毕为止。该模型是增量开发的极端形式，它不仅是增量开发也是增量提交，用户将最早收到部分软件，能及早发现问题，使修改扩充更容易。

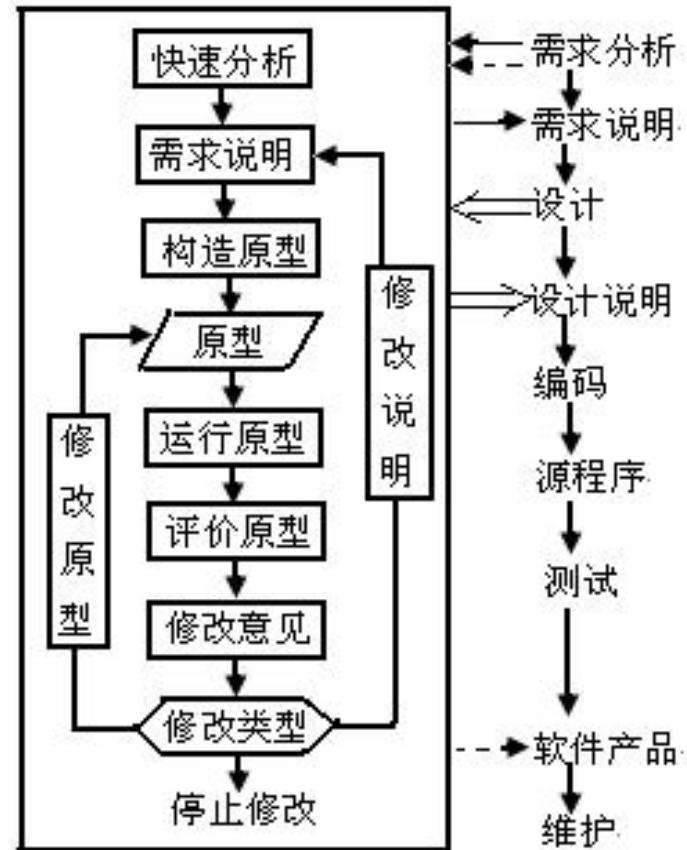
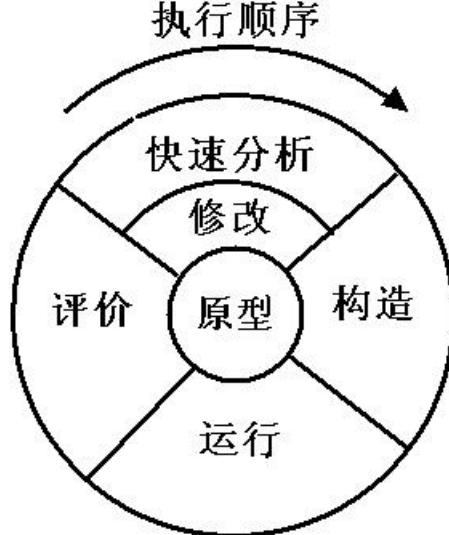
演化提交模型



3. 快速原型模型

软件开发的早期阶段应该是一个学习与实践的过程，其活动包括开发人员和用户两个方面。不仅要求他们合作，而且要有一个实际的工作系统。用户开始虽然说不出未来系统的样子，但对现行系统可以非常熟练的使用。基于这种思想的技术就是快速原型开发。

快速原型模型



2.3.5 螺旋模型

螺旋模型综合了传统的软件生命周期模型和原型模型的优点。将瀑布模型与增量模型结合起来，加入了两种模型均忽略了的风险分析，弥补了这两种模型的不足。

螺旋模型是一种风险驱动的模型。在软件开发中，有各种各样的风险。

计划

风险分析

初始需求分析与项目计划

基于用户说明的计划

用户评价

用户评价

工程

基于初始需求分析风险分析

基于用户反馈的风险分析

决策继续与否

完成系统方向

初始软件模型
第二次模型

工作系统

每个螺旋周期可分为四个工作步骤。

1. 计划

确定软件产品各部分的目标，如性能、功能和适应变化的能力等；确定软件产品各部分实现的各种方案；确定不同方案的限制条件。

2. 风险分析

对各个不同实现方案进行评估，对出现的不确定因素进行风险分析，提出解决风险的策略，建立相应的原型。若原型是可运行的则可作为下一步产品演化的基础。

3. 工程

若以前的原型已解决了所有性能和用户接口风险，而且占主要位置的是程序开发和接口控制风险，那么接下来的应是采用瀑布模型的方法，进行用户需求、软件需求、软件设计和软件实现等。同时要对其作适当修改，以适应增量开发。也可选择原型、模拟原型，导致了步骤的不同子集的使用。

4. 用户评价

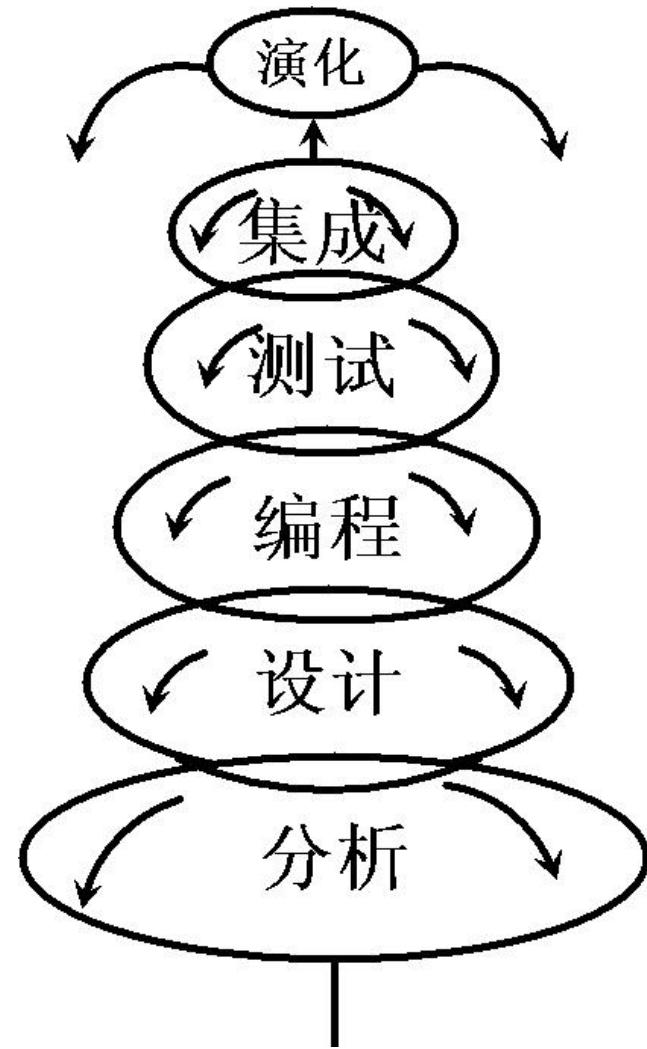
通过对产品的评价，提出修改意见。对下一期的软件需求、设计和实现进行计划。

2.3.6 喷泉模型

软件生命周期模型可以按瀑布模型，先进
行分析，后进行设计；也可以按螺旋模型或增
量模型，交替地进行分析和设计。不过更能体
现两者之间关系特点的是近几年提出的喷泉模
型。

喷泉模型以面向对象的软件开发方法为基
础，以用户需求为动力，以对象作为驱动的模
型。它适合于面向对象的开发方法。

喷泉模型



喷泉模型的特点：

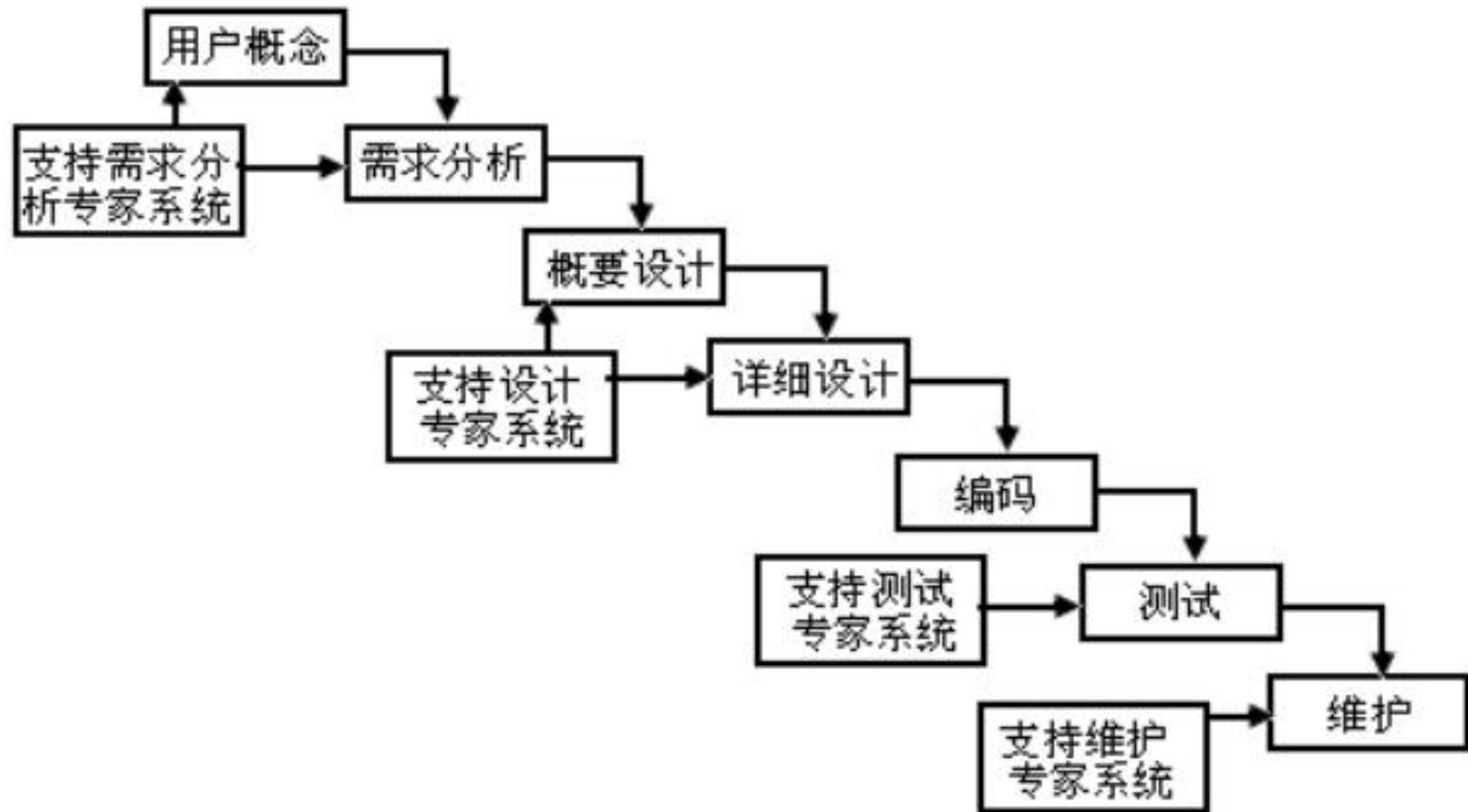
- (1) 模型规定软件开发过程有5个阶段，即分析、设计、实现、测试与集成。
- (2) 模型从高层返回低层无资源消耗，反映了软件过程迭代的自然特性。
- (3) 以分析为基础，资源消耗呈塔型，在分析阶段消耗的资源最多。
- (4) 各阶段相互重叠反映了软件过程并行性。

- (5) 模型强调增量开发，它依据分析一点，设计一点的原则，并不要求一个阶段的彻底完成，整个过程是一个迭代的逐步提炼的过程。
- (6) 模型是对象驱动的过程，对象是所有活动作用的主体和项目管理的基本内容。
- (7) 模型在实现时，由于活动不同，可分为系统实现和对象实现，这既反映了全系统的开发过程，也反映了对象的开发和重用过程。

2.3.7 基于知识的模型

基于知识的模型是把瀑布模型和专家系统结合在一起。在开发的各个阶段上都利用了相应的专家系统来帮助软件人员完成开发工作，使维护在系统需求说明一级上进行。

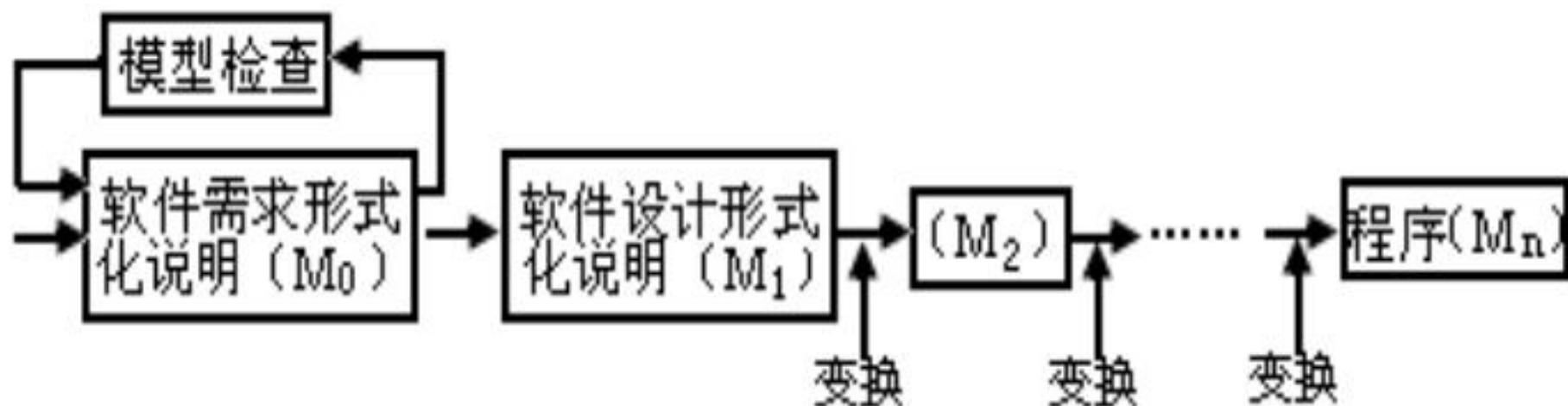
基于知识的模型



基于知识的模型是基于瀑布模型，在各阶段都有相应的专家系统支持。其中支持需求活动的专家系统用来帮助减少需求活动中的二义性、不精确性和冲突易变的需求，这种专家系统要使用应用领域的知识，要用到应用系统中的规则，建立应用领域的专家系统来支持需求活动。

2.3.8 变换模型

变换模型主要用于软件的形式化开发方法，从软件需求形式化说明开始，经过一系列变换，最终得到系统的目標程序。



2.4 软件开发方法

软件开发模型是指开发软件项目的总体过程思路。软件开发方法是一种使用早已定义好的技术集及符号表示习惯组织软件生产过程的方法。

优秀的软件开发方法是克服软件危机的重要途径之一。因此，自软件工程诞生以来，人们重视软件开发方法的研究，已经提出了多种软件开发方法和技术，对软件工程及软件产业的发展起到了不可估量的作用。

2.4.1 结构化方法

结构化方法（Structure Method）是最早的、最传统的软件开发方法。

结构化方法由结构化分析、结构化设计和结构化程序设计构成，也称Yourdon方法。它适用于一般数据处理系统，是一种较流行的软件开发方法。

1. 结构化分析，就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。
2. 结构化设计，就是根据模块独立性准则、软件结构准则，将数据流图转换为软件的体系结构，用软件结构图来建立系统的物理模型，实现系统的概要设计。
3. 结构化程序设计，就是根据结构程序设计原理，将每个模块的功能用相应的标准控制结构表示出来，从而实现详细设计。

2.4.2 Jackson方法

Jackson方法是一种面向数据结构的详细设计方法。

Jackson方法把问题分解为可由三种基本结构形式表示的各部分层次结构。这三种基本结构形式就是顺序、选择和重复。

2.4.3 维也纳开发方法

VDM是一种形式化的开发方法，软件的需求用严格的形式语言描述，把描述模型逐步变换成为目标系统。

VDM是一个基于模型的方法，它的主要思想是：将软件系统当作模型来给予描述，具体说就是把软件的输入/输出看作模型对象，而这些对象在计算机内的状态可看作为该模型在对象上的操作。

2.4.4 面向对象的开发方法

面向对象软件开发 (Objected -Oriented Software Development) 是近年来最流行的软件开发方法。

面向对象方法是一种运用对象、类、继承、封装、聚合、消息传送、多态性等概念来构造系统的软件开发方法。

面向对象开发的基本出发点是尽可能按照人类认识世界的方法和思维方式来分析和解决问题。

面向对象开发方法包括面向对象分析、面向对象设计和面向对象实现。面向对象开发方法有Booch方法、Coad方法和OMT方法等。

2.5 软件工程应用实例

一、背景

教育学及其技术的发展，要求学校对教学成绩管理也要提高质量及效率。因而，需要研制学生成绩管理系统来满足现代教学管理要求。

该产品要充分考虑实际情况。主要完成学生成绩录入、批量导入、学生成绩分析与统计等业务，也可作为学校学生与教务管理系统的子模块。

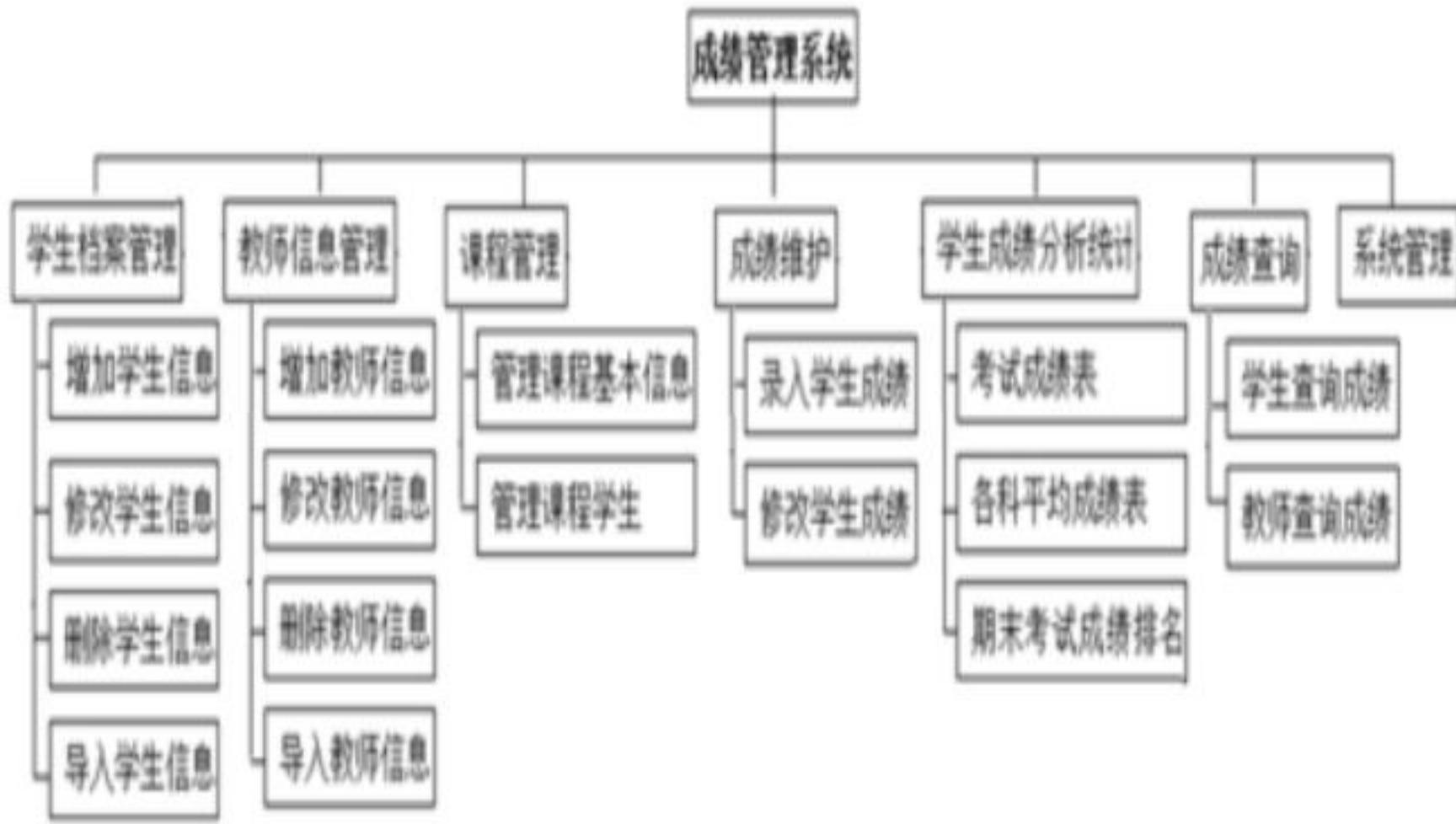
二、目标

要考虑到大学规模，学生人数在5000—50000人。满足600种基本专业（每个专业都有30--40门课程）考试类别多、课程多的特点。对学生成绩的分析统计，适合用于现在的通讯工具。以便于掌握教学情况，改进教学活动。

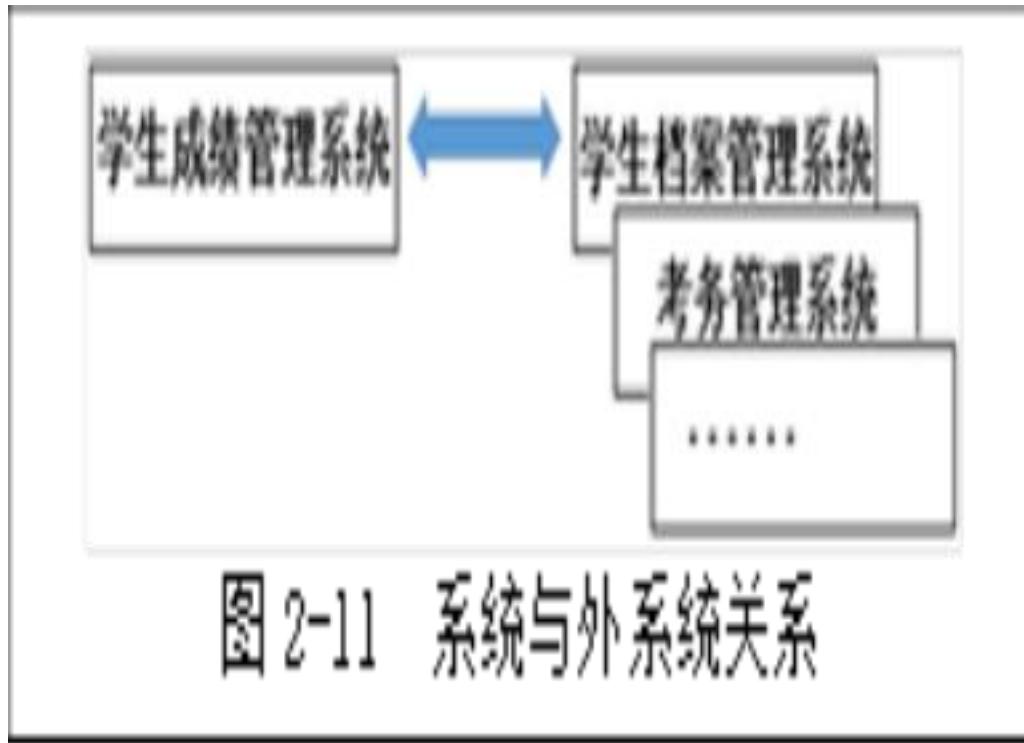
1、用户特点：系统管理员、教师、学生、其他人员。

2、初步功能：用户登陆，退出操作，注册操作。

功能需求



本系统与外系统关系



3、性能需求

- (1) 响应时间不超过1秒；统计时间 ≤ 30 秒。
- (2) 支持20000名学生信息一次性导入时间 ≤ 300 秒。
- (3) 支持5000名用户并发使用，性能不受影响。
- (4) 对每项输出要说明其特征；
- (5) 说明系统的输入数据的组织以及提供的频度；
- (6) 用图表的方式表示出最基本的数据流程和处理流程；
- (7) 本系统相连接的其他系统；

4、安全性需求：权限控制、重要数据加密、备份、记录日志。

5、可用性需求：操作方便、支持无计算机经验、控制必录项、容错能力、操作提示、用户自定义、在线帮助。

6、其它需求：支持多浏览器、安装方便，易于维护。

三、需求规定

1、系统角色：（1）系统管理员、（2）教师、（3）学生。

2、学生管理、教师管理、课程管理、成绩查询、成绩分析与统计。

四、运行环境：单机版和网络环境版；预留用户、硬软件和通信接口。

谢谢