



# 第二章

## 管理信息系统的开发

(Methods of the Management Information System)

[返回总目录](#)



### 教学目的

- 使学生掌握常用的开发方式、策略与方法
- 能够使用一种方法或几种方法进行信息系统的开发

### 教学要求

- 了解管理信息系统开发的方式和策略
- 了解结构化开发方法以及该方法存在的问题，以便更好地进行开发
- 掌握原型法的开发步骤；原型法的种类；原型法的特点及适用范围；
- 掌握面向对象法；面向对象法的开发过程；面向对象法的特点及适用范围
- 具备利用上述方法开发信息系统的能力

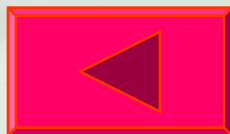
- 管理信息系统开发的方式与策略
- 管理信息开发方法
- 小结



管理信息系统开发概述

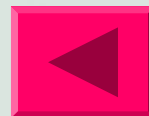
第一节

信息系统开发的方式与策略



### 管理信息系统开发方式

- 管理信息系统开发方式
- 管理信息系统开发结构模式
- 管理信息系统开发的策略
- 管理信息系统开发中存在的问题



### 建立和开发MIS的主要方式

- ◆ 购买现成软件包
- ◆ 自行开发
- ◆ 委托开发
- ◆ 联合开发等





### 信息系统开发方式

方式 特点比较	自行开发	委托开发	联合开发	购买软件包
分析和设计力量的要求	非常需要	不太需要	逐渐培养	少量需要
编程力量的要求	非常需要	不需要	需要	少量需要
系统维护的难易	容易	困难	较容易	困难
开发费用	少	多	较少	较少
说 明	开发时间较长，适用，可以培养自己的系统开发人员	省事，开发费用多。需要业务人员的密切配合	比较适用，用户必须具有一定的人员参加	要有选择，即使符合单位实际，但仍有部分接口问题

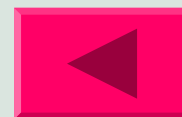


### 开发方式的选择

主要从

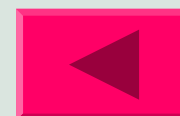
- 用户需求
- 开发成本
- 开发周期
- 技术力量
- 可维护性
- 外部环境等

方面加以比较，选择一种适合本单位的开发方式



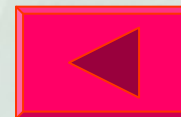
### 开发结构模式的选择

- 随着计算机技术、网络技术的发展，管理信息系统的结构模式也在不断改进
- 主要有五种结构模式
  - 单机模式
  - 中央集中模式
  - 客户机/服务器模式(**Client/Server**, **C/S模式**)
  - 浏览器/服务器模式(**Browser/Server**, **B/S模式**)
  - 多层混合模式。



### 信息系统开发策略

- ❑ “自上而下” (Top--Down) 方法
- ❑ “自下而上” (Bottom--Up) 方法
- ❑ 综合方法





### “自上而下” (Top--Down) 方法

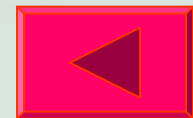
- 从一个组织的**高层管理**着手，考虑组织的目标、对象和策略，确定一个组织的管理信息系统模型
- 确定需要**哪些功能**保证目标的完成，从而划分出相应的**业务子系统**，并进行各子系统的具体分析和设计

### “自上而下”(Top--Down)方法

1. 分析系统**整体**目标、环境、资源和约束条件
2. 确定各项主要业务处理功能和决策功能，从而得到各个**子系统的分工、协调和接口**
3. 确定每一种功能（子系统）所需要的输入、输出、数据存贮等
4. 对各子系统的**功能模块和数据**进行进一步分析与分解
5. 根据需求和可能，确定**优先**开发的子系统及数据存贮等

### “自上而下”（Top--Down）方法

- “自上而下”方法的整体性、逻辑性较强
- 工作量太大而影响具体细节的考虑，致使周期变长
- 开发费用增加
- 评价标准难以确定



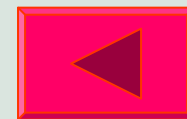


### “自下而上” (Bottom-Up) 方法

- 从各种基本业务和数据处理着手，进行分析和设计
- 当下层子系统分析完成后，再进行上一层系统的分析与设计，将不同的功能和数据综合起来考虑
- 除增添新的功能和数据外，还要考虑一定的经济管理模型

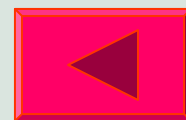
### “自下而上” (Bottom--Up) 方法

- 从具体的业务信息子系统逐层综合再集中到总的管理信息系统的分析和设计，实际上是模块组合的方法
- 可根据资源情况逐步满足用户要求，边实施边见效，但其整体目标和协调性较差
- 可能导致功能及数据的矛盾、冗余，造成返工



### 综合方法

- 首先“**自上而下**”地制定一个组织的**总体方案**
- 再“**自下而上**”地进行具体业务信息系统的**总体设计**
- 这两种方法的结合，可以对系统进行全面的分析，可保证系统的协调和完整，能得到一个比较理想的，耗费人力、物力、时间较少的用户满意的新系统





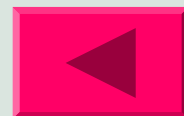
### 信息系统开发中存在的主要问题

- ❑ 认识偏差
- ❑ 管理中应用计算机的基本条件不完全具备
- ❑ 建设没有一整套的规划方案
- ❑ 系统开发中存在一些问题
- ❑ 对管理信息系统开发的关键把握不准
- ❑ 教育、理论体系研究落后
- ❑ 开发方法选择存在问题



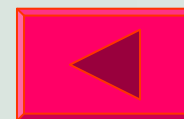
### 认识偏差

- 管理信息系统的建设与评价侧重计算机硬件配置，而不是信息开发与利用的方法与深度
- 组织中信息管理机构的命名突出计算机，而不是信息与信息管理
- 管理信息系统的人才选择与培养强调计算机知识，而不是管理、数学及系统知识
- 管理信息系统的建设缺乏复合型人才，而不是计算机人才



### 管理中应用计算机的基本条件不具备

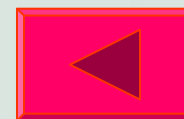
- 企业没有规范化管理——没有科学管理的基础
- 领导不重视，业务人员参与的积极性不高
- 缺少一只开发的专业队伍
- 缺少一定的资源条件





### 建设没有一整套的规划方案

- 管理信息系统建设是一项系统的工程，开发前要有一套行之有效的系统**建设规划方案**
- 规划方案要不断地进行**修改**



### 信息系统开发中存在的主要问题

- 软件开发**生产率**低，跟不上硬件的发展速度

21世纪，计算机硬件的功能将比80年代末提高1000--100000倍，计算机软件的**生产率**及其性能的提高却只有5--100倍”，计算机软件的**生产率**及其性能将大大落后于硬件的发展速度，计算机**软件**已成为计算机技术和应用发展的“瓶颈”

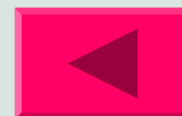
- 软件系统**质量**低，不能满足用户的需求
- 软件开发**成本**高
- 管理软件跟不上发展需要
- 在开发中文档的整理不及时

### 管理信息系统建立过程可能遇到的问题

1. 企业管理制度和运行机制不完善，企业管理不规范
2. 企业的需求不明确，解决方案脱离企业的实际情况
3. 企业缺乏专业化的人才，**MIS**运作不畅

### 管理信息系统建立过程可能遇到的问题

- 4. 企业员工包括领导层对企业的管理信息系统的认同和参与不够
- 5. 企业内部各部门之间缺乏配合，协调不利
- 6. 软件开发合作伙伴选择不当，技术和设备没有及时更新





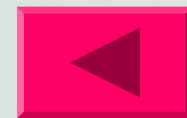
### 对管理信息系统开发的关键把握不准

- “只要熟练掌握几门计算机语言，就可以成为一个优秀的信息系统开发人员”
- 这种观点是极其错误的。
- 计算机程序设计语言是实现计算机信息系统的一种工具或手段，编码只不过是计算机信息系统开发过程中的一小部分工作（约占时间和费用的20%左右）

### 对管理信息系统开发的关键把握不准

建立计算机信息系统主要进行的工作是：

- 要搞清楚系统用户的**基本需求**是什么 (What to do)
- 考虑要怎么做 (How to do )。
- 计算机信息系统开发的**关键**是如何**描述问题**以及如何**解决问题**。



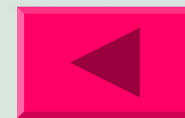
### 教育、理论体系研究落后

- 在教育方面主要表现在**教学内容陈旧**，**理论落后**于实践，理论在某种程度上又**脱离实践**，在教学中往往注重学生的**编程技巧**能力培养，而**忽视**系统分析、设计能力的培养，学生的实践能力**差**，团队合作能力**差**
- 系统开发本身还缺乏一套严格的**理论基础**以及缺少一套简单有力的**开发工具**



### 开发方法选择存在问题

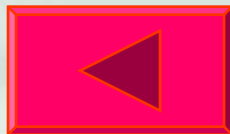
- 传统的软件开发方法不允许在开发过程中用户的需求发生变化，需求分析不彻底仍是导致应用开发软件失败的主要原因之一
- 企业管理信息系统建设中选用的开发工具和开发方法是否恰当，直接影响着建设的质量和速度





## 第二节

# 管理信息系统开发方法



### 管理信息系统开发方法

- ❑ 结构化生命周期法
- ❑ 快速原型法
- ❑ 面向对象法
- ❑ 系统开发方法的选择



### 教学目的

- 使学生掌握常用的结构化开发方法
- 结构化方法是常用的信息系统开发方法，是其他方法学习的基础，必须掌握其使用

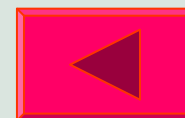
### 教学要求

- 了解结构化开发方法以及该方法存在的问题
- 具备利用结构化方法开发信息系统的能力



### 结构化开发方法

- 结构化方法的由来
- 结构化方法的基本思想
- 结构化系统开发的基本原则
- 结构化开发方法的一般过程
- 结构化开发方法的特点

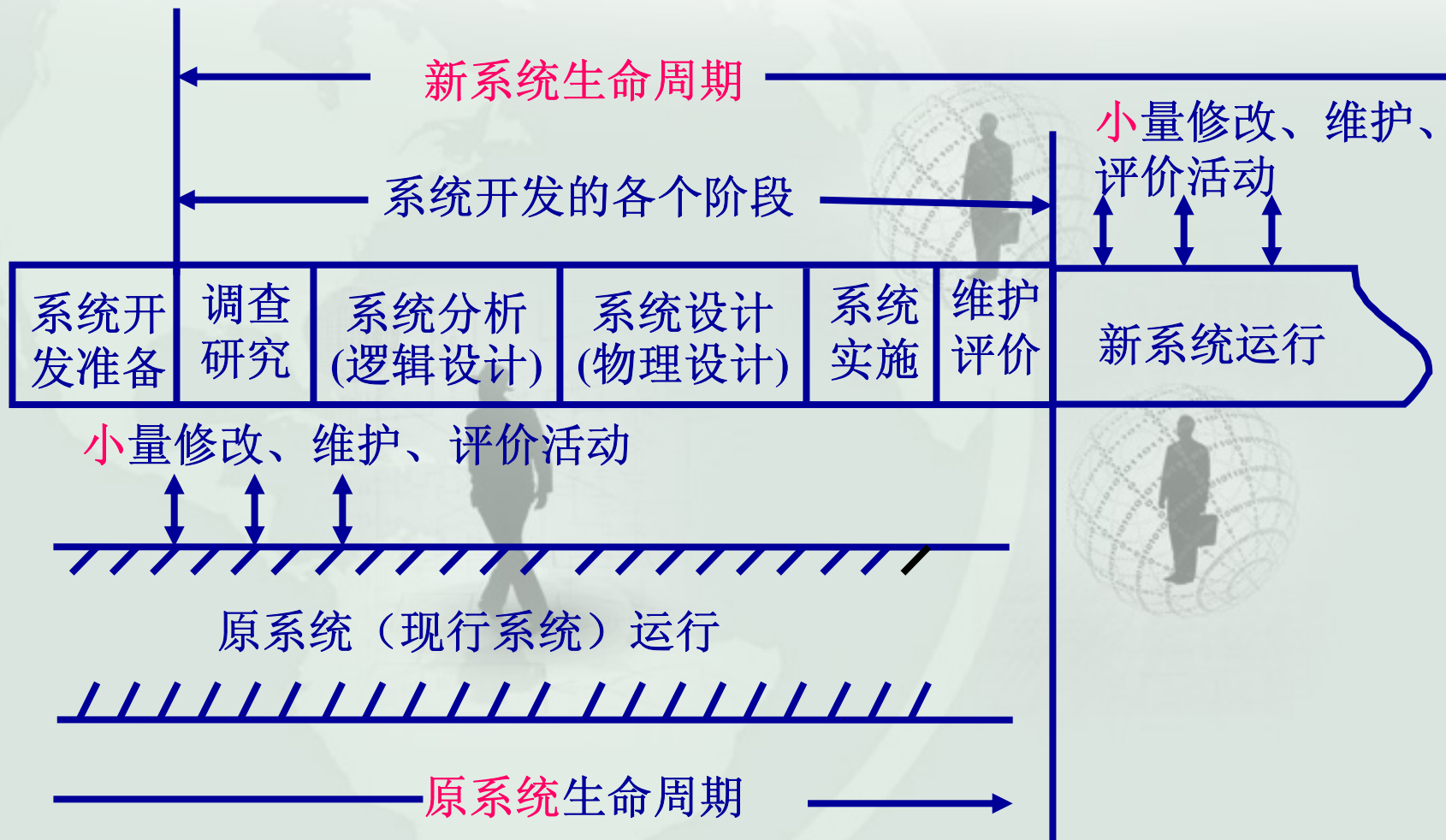


### 结构化开发方法的由来

系统的生命周期：

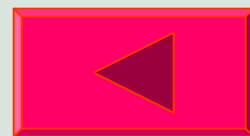
任何一个系统都有产生、发展和消亡的过程，新系统是在旧系统的基础上产生、发展、老化、淘汰，最后又被更新的系统所取代

### 结构化开发方法的由来



### 结构化开发方法的由来

- 20世纪70年代，西方国家逐渐发展了结构化方法
- 结构化的概念最早用来描述结构化程序设计方法
- 结构化的程序设计思想引入到系统分析与设计中，形成了结构化的分析与设计的方法
- 生命期中应用结构化的理论，就是结构化的生命周期方法，简称为结构化方法

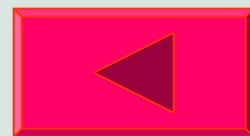




### 结构化开发方法——基本思想

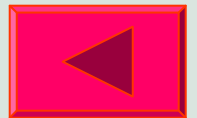
结构化方法的基本思想：

用系统的思想，系统工程的方法，按**用户至上**的原则，**结构化、模块化、自顶向下**地对信息系统进行分析与设计



### 信息系统的开发原则

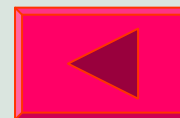
- 面向用户的观点
- 严格区分工作阶段，每个阶段规定明确的任务和所应得的成果
- 按照系统的观点，自顶向下地完成研制工作
- 充分考虑变化的情况
- 工作成果规范化、标准化



### 信息系统的开发步骤

- ①总体规划与可行性分析    ②信息系统分析
- ③信息系统设计              ④信息系统实施
- ⑤信息系统运行管理与评价

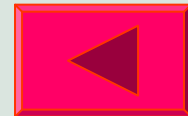
信息系统开发阶段



### 总体规划与可行性分析

#### 1. 任务:

- 对现行系统进行初步调查研究
- 进行系统的规划，提出新系统方案
- 对新系统方案进行可行性研究
- 撰写可行性分析报告



### 系统总体规划与可行性分析

#### 系统规划的内容:

- (1) 从总体的角度来规划系统应该由哪些部分组成
- (2) 在这些组成部分中有哪些数据类, 它们之间如何交换数据
- (3) 根据信息与功能需求提出计算机系统硬件、网络配置方案
- (4) 根据管理需求来确定这些模块的开发优先顺序
- (5) 制定出开发计划, 根据开发计划合理调配人员、物资和资金

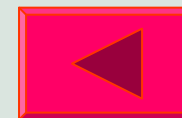


### 总体规划与可行性分析

#### 2. 可行性分析的目的

避免盲目投资，减少不必要的损失

#### 3. 成果——可行性研究报告



### 信息系统分析

#### 1. 任务:

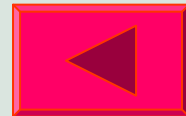
- (1) 对系统组织机构、业务流程进行详细调查，获取系统的信息需求
- (2) 详细分析系统的业务流程，用业务流程图表示
- (3) 从业务流程抽取出数据流程，用数据流程图表示
- (4) 构建新系统的逻辑模型
- (5) 撰写系统分析报告

### 信息系统分析

## 2. 系统分析结果

系统分析进行完之后，要写出**系统分析报告**

- 与用户进行充分地交流和验证, 检验正确后可进入下一阶段工作



### 系统设计

#### 1. 系统设计的任务：

##### (1) 系统的功能结构设计

结合计算机的具体实现, 设计各个组成部分在计算机系统上的结构

即采用一定的**标准**和**准则**,

考虑模块应该有哪些

程序块组成,

它们之间如何联系

##### (2) 系统的**处理流程**设计

##### (3) 系统的**编码**设计

##### (4) 数据**文件**和**数据库**设计

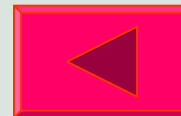
##### (5) 系统的**输入/输出**设计

##### (6) **网络**设计

### 系统设计

#### 2. 系统设计的成果

系统设计阶段的成果是系统设计说明书





### 系统实施

#### 1. 任务:

(1) 系统硬件设备的购置与安装

(2) 程序的编制

(3) 数据的录入设计是根据系统设计阶段的成果，  
遵循一定的设计原则来进行的。

(4) 人员的培训

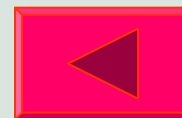
(5) 系统的测试、调试和转换

#### 2. 系统开发实施的成果:

(1) 程序清单

(2) 测试与调试报告

(3) 系统使用说明书



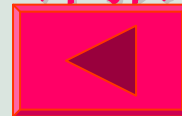
### 信息系统运行管理与评价

#### 1. 任务:

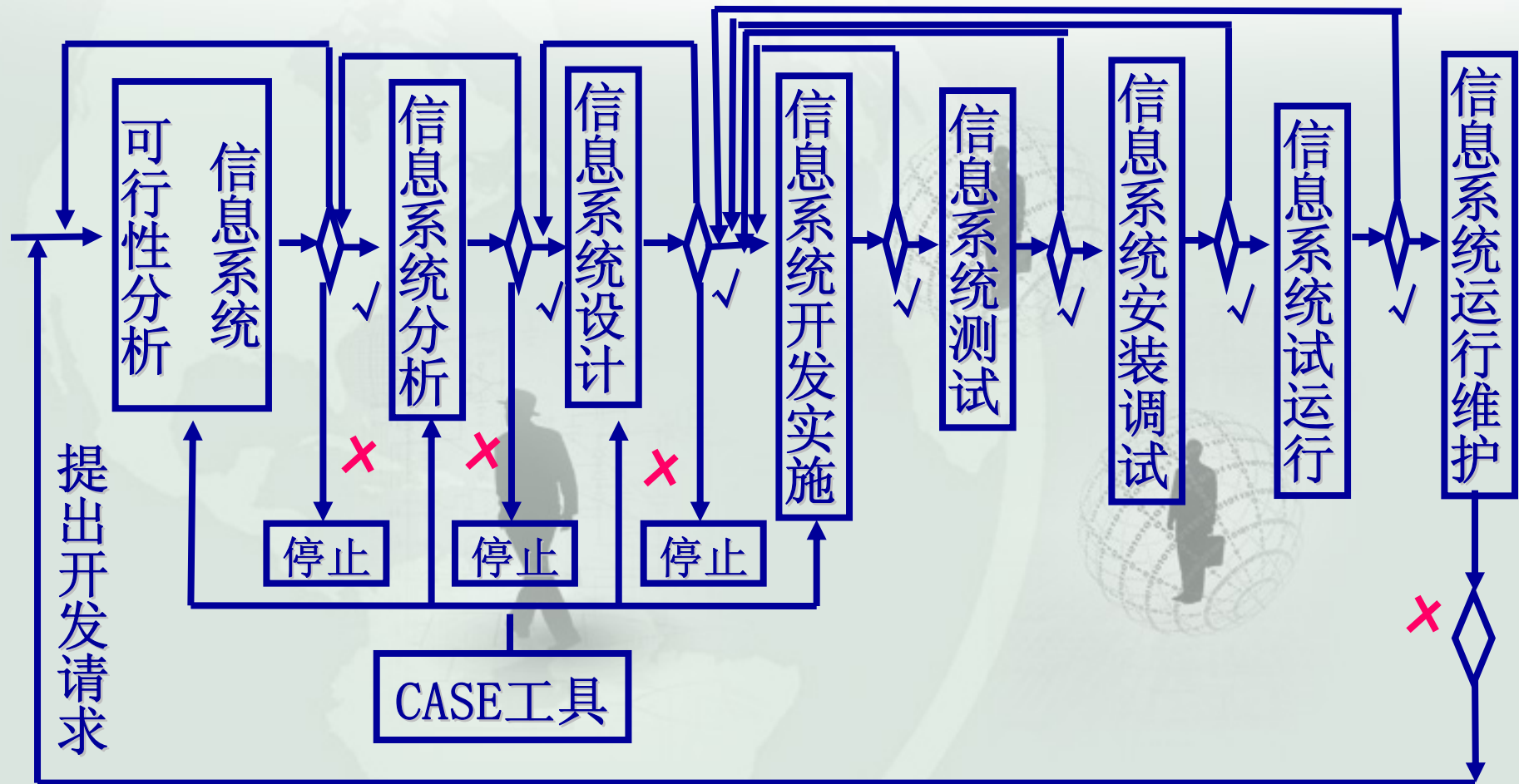
- 对系统进行修改与扩充（即系统维护）
- 日常运行管理
- 运行情况的记录
- 对系统运行情况进行检查与评价
- 撰写出系统评价报告

#### 2. 成果

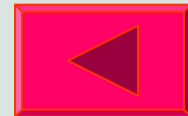
系统运行维护阶段的成果是系统运行审计报告



### 信息系统开发过程



信息系统开发阶段

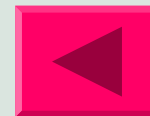


### 结构化方法的特点与局限性

- 结构化生命周期法的假设是**预先定义**需求的策略
- 使用生命周期法，项目的参与者之间往往存在**通信鸿沟**
- 生命周期法使用的基本技术，是结构化分析（SA）和结构化设计（SP）技术，结构化和结构设计是建立在**系统生命周期**的概念基础上的
- 用户的需求往往造成系统结构较大的变化，维护困难
- 系统分析和系统设计阶段所采用的**工具不同**
- 过于耗费资源

### 结构化方法的特点与局限性

- 用这种方法开发的软件，其稳定性、可修改性和可重用性比较差
- 结构化生命周期法特别适合于开发那些能够预先定义需求、结构化程度又比较高的大型事务型系统（TPS）和管理信息系统（MIS）
- 生命周期法不适合于开发信息需求不明确的系统





### 教学目的

- 使学生掌握常用的快速原型法开发方法
- 能够使用快速原型法进行信息系统的开发

### 教学要求

- 掌握原型法的开发步骤；
- 原型法的种类；
- 原型法的特点及适用范围；
- 具备利用原型化法开发信息系统的能力

### 快速原型法

- ❑ 原型法概述
- ❑ 原型法方法的开发过程
- ❑ 原型法的种类和构造过程
- ❑ 原型法的特点
- ❑ 原型法的使用场合与局限性



### 原型法概述

- ❖ 系统的种类
- ❖ 原型法的由来
- ❖ 原型法的基本思想



### 系统的种类

#### (1) 预先指定的系统

系统的需求比较稳定而且能够预先指定

例如，传统工业生产过程的计算机控制系统，卫星图象处理系统，空中交通管理系统，火箭发射跟踪系统，操作系统，编译程序、DBMS类的软件。



### 系统的种类

#### (2) 用户驱动系统

系统的需求使模糊的或随时间变化的，通常在安装运行之后，还会由用户驱动对需求进行动态修改。



### 原型法的由来

- 20世纪60~70年代初期，管理信息系统使用的范围较窄，使用环境较稳定
- 随着人们对管理信息的了解，应用越来越广泛，有些系统的需求是不稳定或经常变动的
- 在70年代中期，人们提出了旨在改进SDLC缺点的一种开放式方法——原型法（Prototyping）



### 原型法的基本思想

- 原型法就是根据用户提出的**需求**，由用户与开发者共同确定系统的**基本要求**和**主要功能**，并在较短时间内建立一个**实验性的、简单**的小型系统，称做“原型”，然后将原型交给用户使用
- 用户在使用原型的过程中会产生**新的需求**，开发人员依据用户提出的评价意见对简易原型进行不断的**修改、补充**和**完善**。如此不断地反复修改、直至满足用户的需求
- 形成了一个相对**稳定**、较为**理想**的管理信息系统



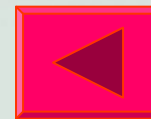
### 原型法的开发过程

- (1) 识别基本需求
- (2) 构造初始原型
- (3) 使用和验证原型
- (4) 修正和改进原型
- (5) 判断原型完成
- (6) 生成文档并交付使用



### 识别基本需求

- 系统结构
- 功能
- 输入的要求
- 输出的要求
- 数据库基本结构
- 系统接口



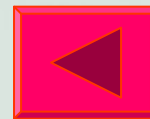


### 构造工作原型

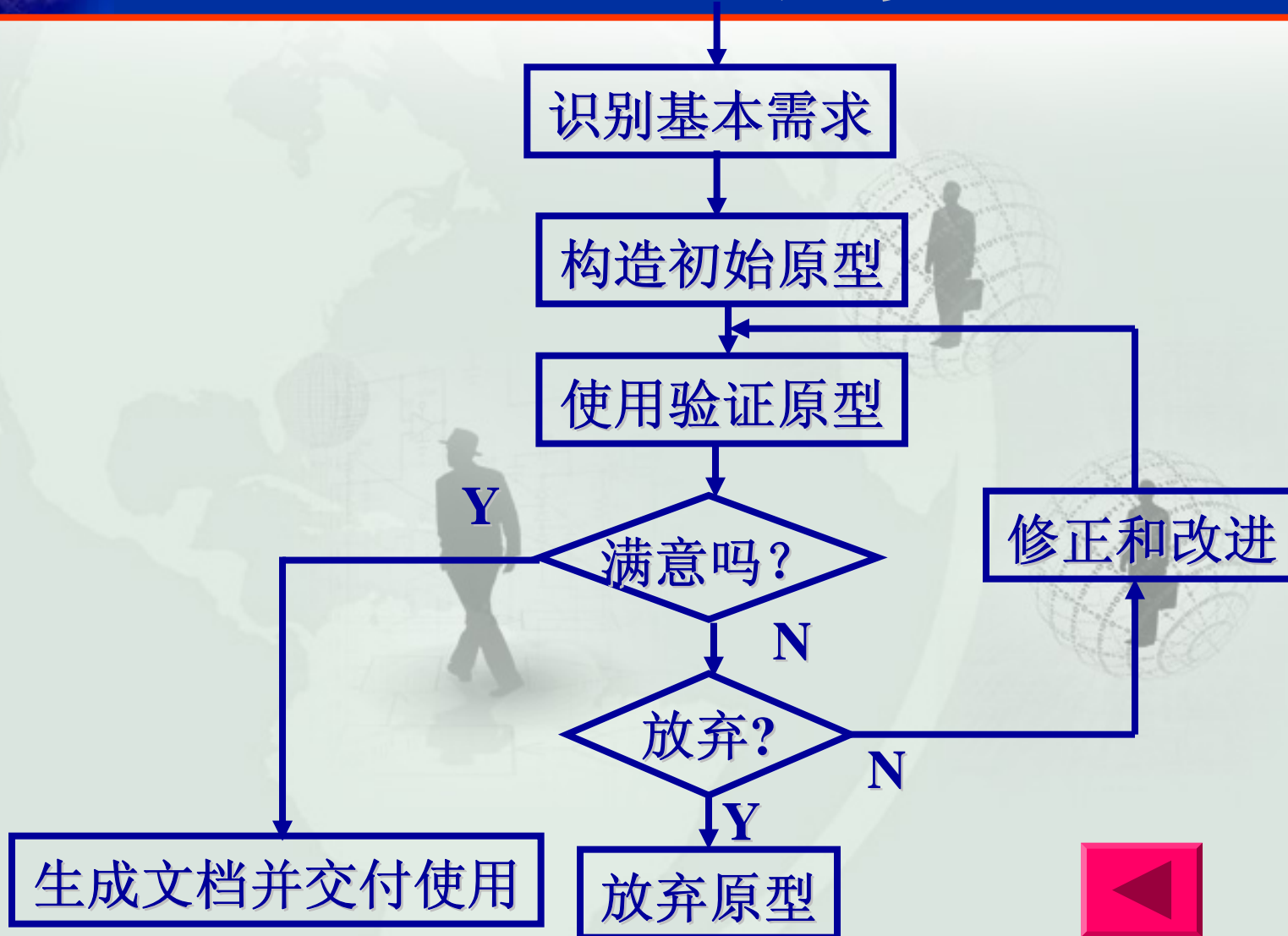
- 根据系统的基本需求建立**原型**的初始方案
- 对初始方案进行**讨论**和**迭代**
- 工作模型包括**数据库**模型和系统**功能**模型
- 初始模型的**质量**对生成新的信息系统至关重要

### 迭 代

- 迭代就是用户对原型系统进行评价后，提出意见，开发人员根据用户的意见进行修改的反复过程。
- 迭代不是简单的反复，每一次迭代都意味着原型系统向着用户需求又前进了一步，迭代是系统开发进展的动力。迭代过程的结束，就是原型的完成。



### 原型法的开发步骤



### 原型的种类和构造方法

❖ 原型的种类

❖ 原型的构造方法



### 原型的种类

根据它在开发过程中的作用

- 丢弃式原型
- 演进式原型

根据原型的目的和应用场合

- 研究型原型
- 实验型原型
- 演进型原型

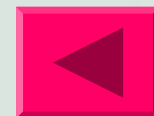




### 丢弃式原型

#### 丢弃式（Throw-it-away Prototyping）原型

- 在于描述和说明系统的需求作为开发人员和用户之间的通信工具
- 不作为实际系统运行
- 原始系统只是从外观上、功能上象“实际系统”



### 演进式原型

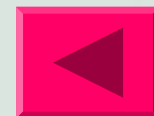
#### 演进式（Evolutionary Prototyping）原型

- 进化式原型开发思想与丢弃式完全相反
- 按照基本需求开发出一个系统，让用户先使用起来，随时修改
- 原系统的一部分或几部分不适用，需要重新设计、实施和安装



### 根据原型的目的和应用场合划分

- **研究型：**主要针对系统目标模型，用户及开发者缺乏经验，目的是弄清系统目标
- **实验型：**用于大规模开发和实现之前的考核、验证方案是否合适，系统说明是否可靠
- **演进型：**目的是在改进过程中对原型进行积累和扬弃，并将原型演进成为最终的系统



### 演进型原型构造方法

➤ 演变式 (Spiral Prototyping) 系统开发

➤ 递增式 (Incremental Prototyping) 系统开发



### 演化式系统开发 (Spiral prototyping)

- ✓ 演化式系统开发的原理
- ✓ 演化式系统的开发过程





### 演化式原型的原则

- 演化式系统开发思想与**丢弃式**原型完全**相反**
- 把系统开发看出一种周期的过程，从设计到实现，再到评估，反复进行
- 最终产品被看成是一种各个阶段评估的**版本序列**
- 研究型 and 实验型原型构造模式可以在演变式开发的早期使用





### 递增式系统开发 (Incremental prototyping)

- 递增式系统开发原理
- 递增式系统开发的过程



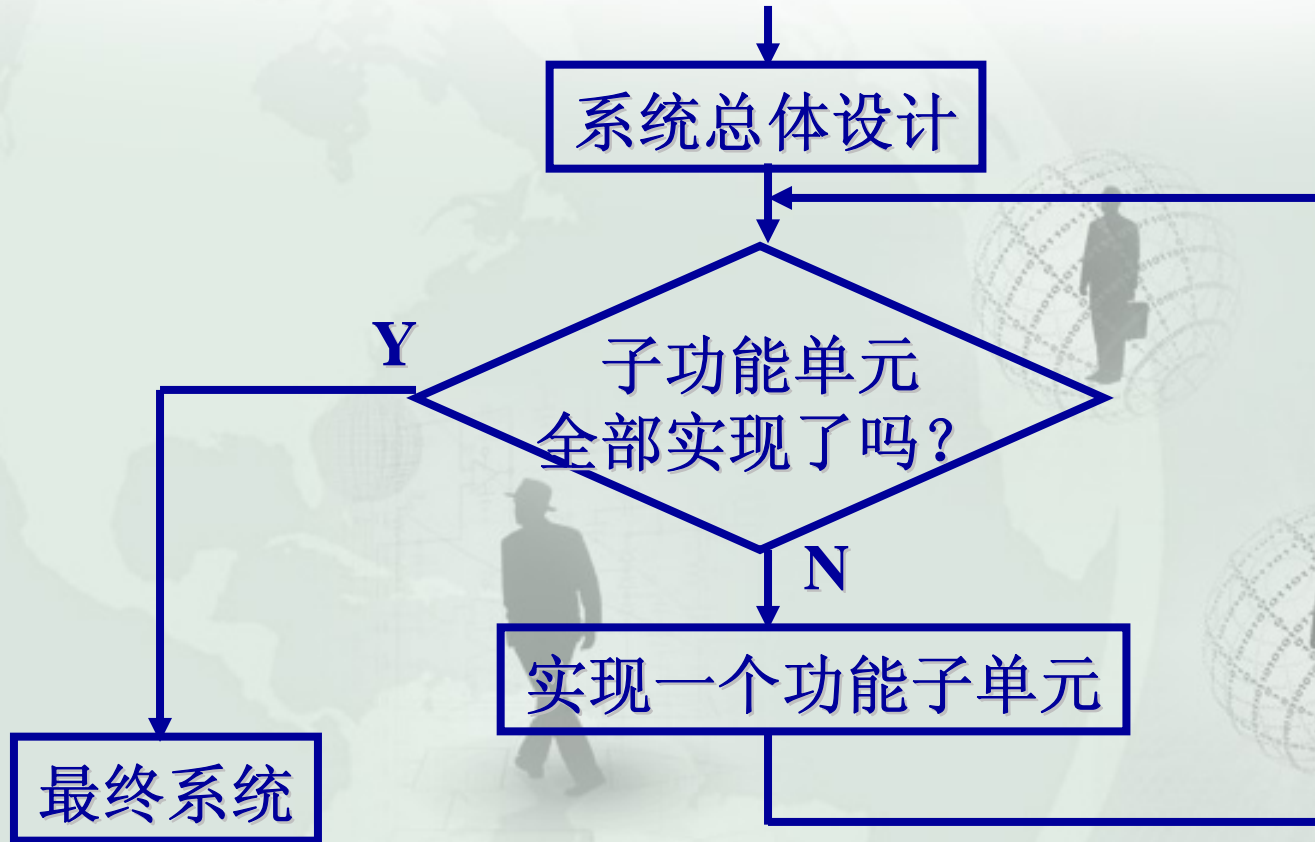
### 递增式系统开发的原理

- 开始时系统有一个**总体框架**，各功能单元的结构和功能十分清楚
- 功能还**没有**具体**实施**
- 以后开发中逐步**完善模块**

类似于计算机工业中的  
插接策略 (**plug-in strategy**)



### 递增式系统开发过程



系统是在原型的基础上逐步扩展，横向发展起来的





### 原型法的特点

- 系统开发效益高
- 系统适用性强
- 系统可维护性好
- 系统可扩充性好
- 易学易用
- 原型化方法缺乏统一的规划和开发标准，难以对开发过程进行控制
- 对环境要求高
- 用户往往出现急躁情绪
- 常常用原型代替系统分析

### 原型法的使用场合与局限性

- 原型化方法适用于用户需求不清，管理及业务处理不稳定，需求常常变化
- 规模小，不太复杂，而且不要求集中处理的系统
- 有比较成熟借鉴经验的系统开发
- 用于开发信息系统中的最终用户界面
- 原型法的最大优点是能提高用户满意度
- 使用原型法开发系统周期短，成本低

### 原型法的使用场合与局限性

- 原型化方法不适于开发大的系统
- 原型化方法开发的时候，测试和文档工作常常容易被忽略
- 原型化方法的另一个缺点是运行的效率可能会比较低



### 教学目的

- 使学生掌握常用的面向对象**开发方法**
- 能够使用面向对象方法进行信息系统的开发

### 教学要求

- 掌握面向对象法的种类;
- 掌握面向对象法的开发过程;
- 掌握面向对象法的特点及适用范围
- 具备利用面向对象方法开发信息系统的能力



### 面向对象方法

- ❑ 面向对象方法的产生和发展
- ❑ 面向对象方法通用原则
- ❑ 面向对象的概念
- ❑ 面向对象方法的种类
- ❑ 面向对象方法的开发过程
- ❑ 统一建模语言
- ❑ 面向对象方法的特点

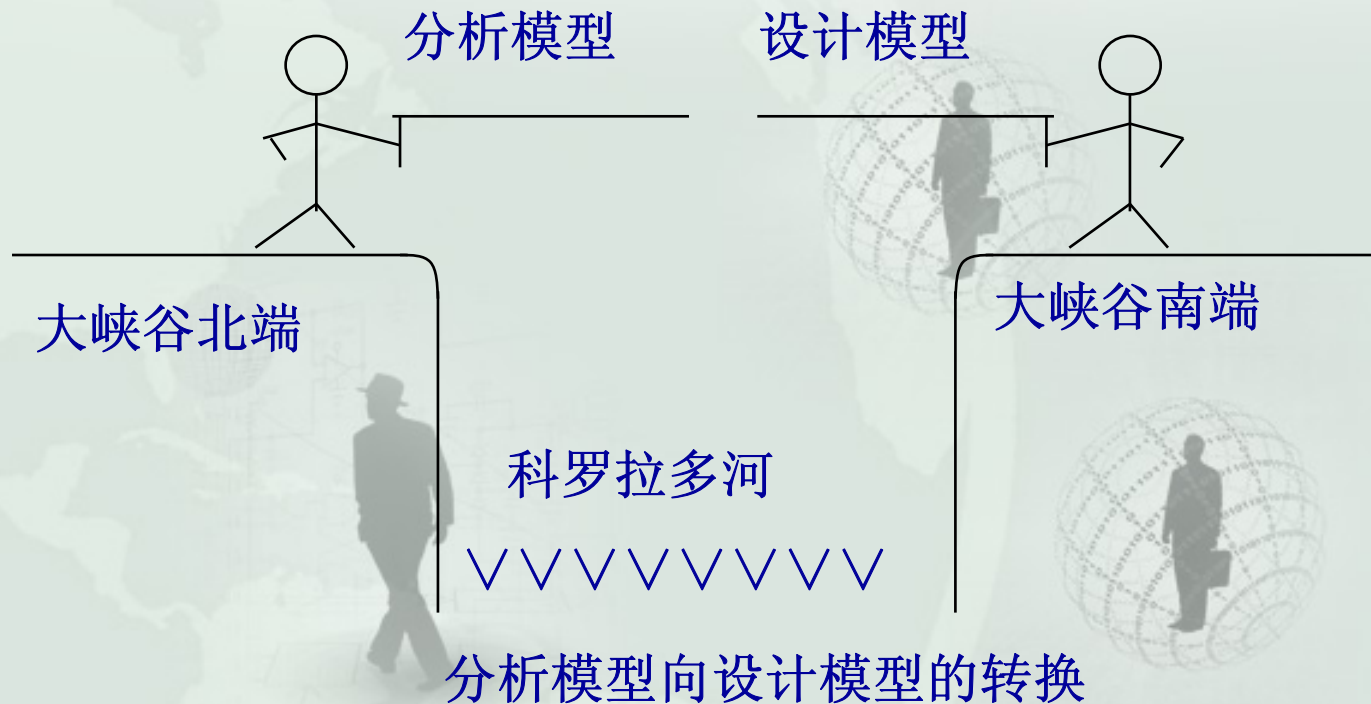


### 面向对象方法的产生与发展

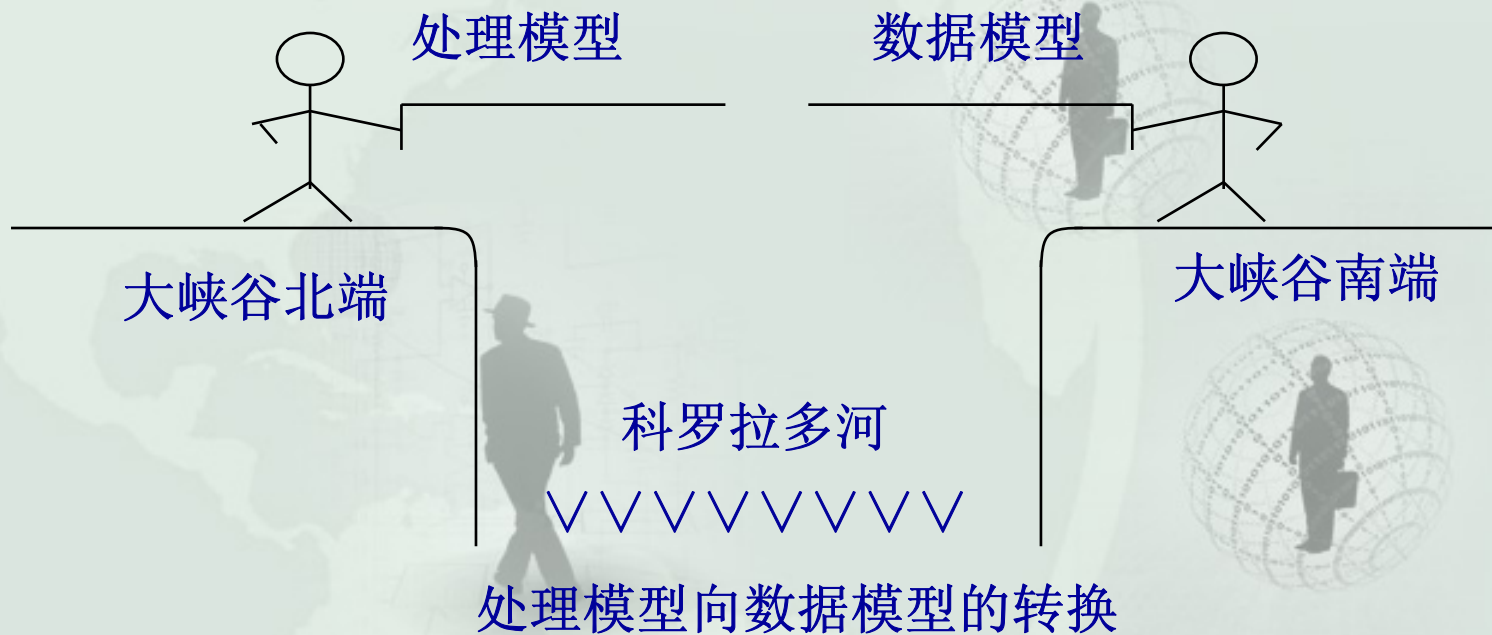
#### 传统开发方法存在的问题

- 问题空间与求解空间的**不一致**
- 系统分析到系统设计**转换**困难
- **处理**模型和**数据**模型分别建立
- 忽视信息系统的行为特征

### 面向对象方法的产生与发展



### 面向对象方法的产生与发展



### 面向对象方法的产生与发展

#### 面向对象方法的由来和历史

- 面向对象（Object Oriented，简称OO）的概念应起源于挪威的K . Nyguard等人开发的模拟离散事件的程序设计语言Simula 67
- 真正的面向对象程序设计（Object Oriented Programming，简称OOP）是由Alan Keyz主持设计的Smalltalk语言
- 20世纪80年代面向对象的分析、面向对象的设计等方法和技术才开始兴起



### 面向对象方法的产生与发展

- 1988年Shlaer和 Mellof 首先在其《面向对象的系统分析》（Object-Oriented System Analysis）一书中，集中使用**ER模型**（ Entity-Relationship Model）来捕捉用户需求信息
- 1991年 Coad和Yourdon在《面向对象的分析》一书中提出了OOA面向对象分析和OOD面向对象设计方法
- 在1991年，Rumbaugh等人提出了一个用于系统分析和设计的“**面向对象的建模技术**（Object-Oriented Modeling Technology，简称为OMT）”
- 1992年，Embley等人又提出了OSA（面向对象的系统分析）方法

◆ 抽象

## ❖ 封装与信息隐蔽

## ◆ 继承

继承是表示相似性质的机制

◆ 多态

## 多态一般指具有多种形态的能力

◆ 消息

## 消息通信是面向对象的方法中对象之间相互联

## ◆ 关联

复用就是重复使用。复用可以采用3共享、拷贝和改造

## ◆ 复用

### 通用组织方法

对象及其属性或特征

整体和部分 使用“有一 (has a)”逻辑

类和成员 使用“是一个 (is a) 逻辑



### 面向对象的概念——对象

- **对象**是面向对象系统运行过程中的基本**实体**
- 既包括**属性**，也包括作用于属性的**行为**
- 传统语言的操作数据是**被动**地等待相应的操作去处理，但对象是一个**主动**的数据实体，其中**封装**了一组对该数据的**操作**，所以对“对象”和传统的“数据”的使用方法也不同

### 面向对象的概念——消息

- 消息是对象之间进行通讯的一种**数据结构**
- 对象之间是通过传递消息来进行**联系**的
- 消息用来请求对象执行某一处理或提供某些信息的要求，**控制流**和**数据流**统一包含在消息中
- **程序**的执行是靠对象间传递消息来连接的



### 面向对象的概念——消息

#### 消息传送与函数调用的区别

- 函数调用可以带或不带参数，但是消息至少带一个参数，它表明接收消息的对象
- 消息操作名类似于函数名，但它们之间的本质区别在于：函数名代表一段可执行的代码，但消息操作名具体功能的选取还取决于接收消息的对象本身
- 函数调用是过程式的（如何做），消息传送是说明式的（做什么），具体如何做，由对象根据收到的消息自行确定

### 面向对象的概念——类

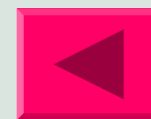
- 类是对一**组**对象的抽象
- 将该组对象所具有的**共同特征**（包括操作特征和存储特征）集中起来，以说明该组对象的能力和性质

### 面向对象的概念——继承

- 继承指对象**继承**它所在类的**结构、操作和约束**
- 也指一个类**继承超类或子类**继承父类的**结构、操作或约束**。继承是自动**共享**类、子类和对象中的**方法和数据的机制**

### 面向对象的概念——封装

- 封装是一种信息**隐藏**技术
- 所谓“封装”是指把对象的属性和操作方法同时封装在定义对象中
- 用**操作集**来描述可见的模块外部接口，从而保证了对象的界面独立于对象的内部表达
- 对象的操作方法和结构是不可见的，**接口**是作用于对象上的操作集的说明，这是对象唯一的可见部分



### 面向对象的概念——实例

- 实例就是由某个特定的类所描述的一个具体的对象
- 类是对具有相同属性和行为的一组相似的对象抽象
- 类在现实世界中并不能真正存在





### 面向对象的概念——多态

- 所谓多态即一个名字具有多种语义
- 对象的多态是指在一般类中定义的属性或服务被特定类继承之后，可以具有不同的数据类型或表现不同的行为
- 多态引用表示可引用多个类的实例

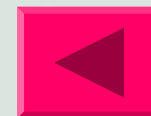


### 面向对象的开发的种类

面向对象的开发方法是面向对象的**分析**、面向对象的**设计**以及面向对象**编程**的总称

它是基于问题对象的自底向上的一种系统开发方法

- ✓ Booch方法
- ✓ Coad\_Youdon方法
- ✓ OMT方法
- ✓ Jacobson方法



### Booch方法

- 在给定的抽象层次上识别类和对象;
- 识别这些对象和类的语义;
- 识别这些类和对象之间的关系;
- 实现类和对象

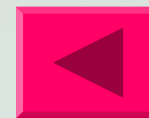


### Coad--Yourdon方法

Coad--Yourdon方法概述

Coad--Yourdon方法的开发步骤

Coad--Yourdon方法的特点



### Coad--Yourdon方法概述

- Coad--Yourdon是美国大学的教授，他俩于1991年合写了一本书《面向对象的分析》
- 该书详细地阐述了面向对象系统分析的一套使用方法和具体步骤
- 用实例进行说明
- 后来他俩又合写了一本书《面向对象的设计》
- 面向对象的分析和设计采用相同的模型，避免了分析和设计的转换





### Coad--Yourdon方法的开发步骤

- 标识对象
- 标识结构
- 定义主题
- 定义属性
- 定义服务

### Coad--Yourdon方法的开发步骤

- **问题域部分**（PDC）——面向对象分析的结果直接放入该部分

- **人机交互部分**（HIC）

对用户分类，描述人机交互的脚本，设计命令层次结构，设计详细的交互，生成用户界面的原型，定义HIC类

- **任务管理部分**（TMC）

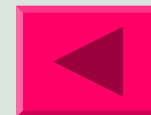
识别任务（进程）、任务所提供的服务、任务的优先级、进程是事件驱动还是时钟驱动、以及任务与其它进程和外界如何通信

- **数据管理部分**（DMC）依赖于存储技术是文件系统、关系数据库管理系统，还是面向对象数据库管理系统



### Coad--Yourdon方法的特点

- Coad/Yourdon方法，是最早的面向对象的分析和设计方法之一
- 简单、易学
- 适合于面向对象技术的初学者使用
- 由于该方法在处理能力方面的局限，目前已很少使用



### OMT（面向对象的建模技术）方法

- OMT方法概述
- 用OMT方法进行分析的步骤
- 用OMT方法进行设计的步骤
- 用OMT方法进行实施的步骤
- OMT方法的特点



### OMT方法概述

- 1991年Rumbaugh等人提出了“面向对象的建模技术（OMT）”
- OMT用于系统分析和设计
- 用OMT方法进行分析，通常要建立系统的三种模型





### 用OMT方法进行系统分析的开发步骤

- 建立对象模型
- 建立动态模型
- 建立功能模型



### 建立对象模型

- 对象模型描述系统中的对象、对象之间的关系
- 标识类中的属性和操作
- 反映系统的静态结构



### 建立动态模型

- 通过对事件和状态的描述和组织构成状态图
- 动态模型表述系统与时间的变化有关的性质



### 建立功能模型

- 功能模型由多张数据流程图组成
- 功能模型描述系统中所有的计算



### 用OMT方法进行系统设计的开发步骤

- 把系统**分解**成子系统
- 识别问题中固有的**并发性**
- 把子系统**分配**给处理器和任务
- 选择**数据存储**管理的方法
- 处理访问全局**资源**
- 选择软件中的**控制实现**
- 处理**边界条件**
- 设置权衡的**优先权**



### 用OMT方法进行对象设计的开发步骤

- 组合三种模型——获得类上的操作
- 实现操作的算法设计
- 优化数据的访问路径
- 实现外部交互式的控制
- 调整类结构提高继承性
- 设计关联
- 确定对象表示
- 把类和关联封装成模块



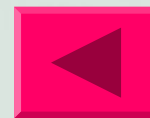
### 用OMT方法进行实现

- 可以使用面向对象的语言
- 非面向对象语言等程序设计语言实现
- 也可以使用数据库管理系统实现



### OMT方法的特点

- OMT被认为是最精确的方法
- 从系统分析到程序设计都能给以详细说明
- 容易建立接近现实的模型
- 比较难掌握，难理解

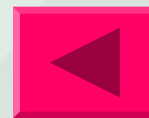


### Jacobson方法

- 面向用例 (Use-Case)
- 比较适合支持商业工程和需求分析
- 需求分析阶段定义潜在的角色和识别问题域中的对象和关系
- 该方法最重要的概念是use case
- 将Use-case模型与5中模型关联
  - ◆ 领域对象模型
  - ◆ 分析模型
  - ◆ 设计模型
  - ◆ 实现模型
  - ◆ 测试模型

### 面向对象方法的开发过程

- 系统调查 (System Diagnoses)
- 面向对象系统分析 (Object-Oriented Analysis)
- 面向对象设计 (Object-Oriented Design)
- 面向对象编程和实现 (Object-Oriented Program)



### 统一建模语言 (UML)

- UML (Unified Modeling Language) 是软件界第一个统一的建模语言
- 该方法结合了Booch, OMT, 和OOSE方法的优点, 统一了符号体系, 并从其它的方法和工程实践中吸收了许多经过实际检验的概念和技术。
- 它是一种标准的表示, 已成为国际软件界广泛承认的标准。是一种基于面向对象的可视化的通用(General)建模语言。为不同领域的用户提供了统一的交流标准 — UML图
- UML应用领域可用于软件开发建模的各个阶段, 商业建模 (Business Modeling), 也可用于其它类型的系统。



### 统一建模语言 (UML)

九十年代中，面向对象方法成为软件分析和设计方法的主流。

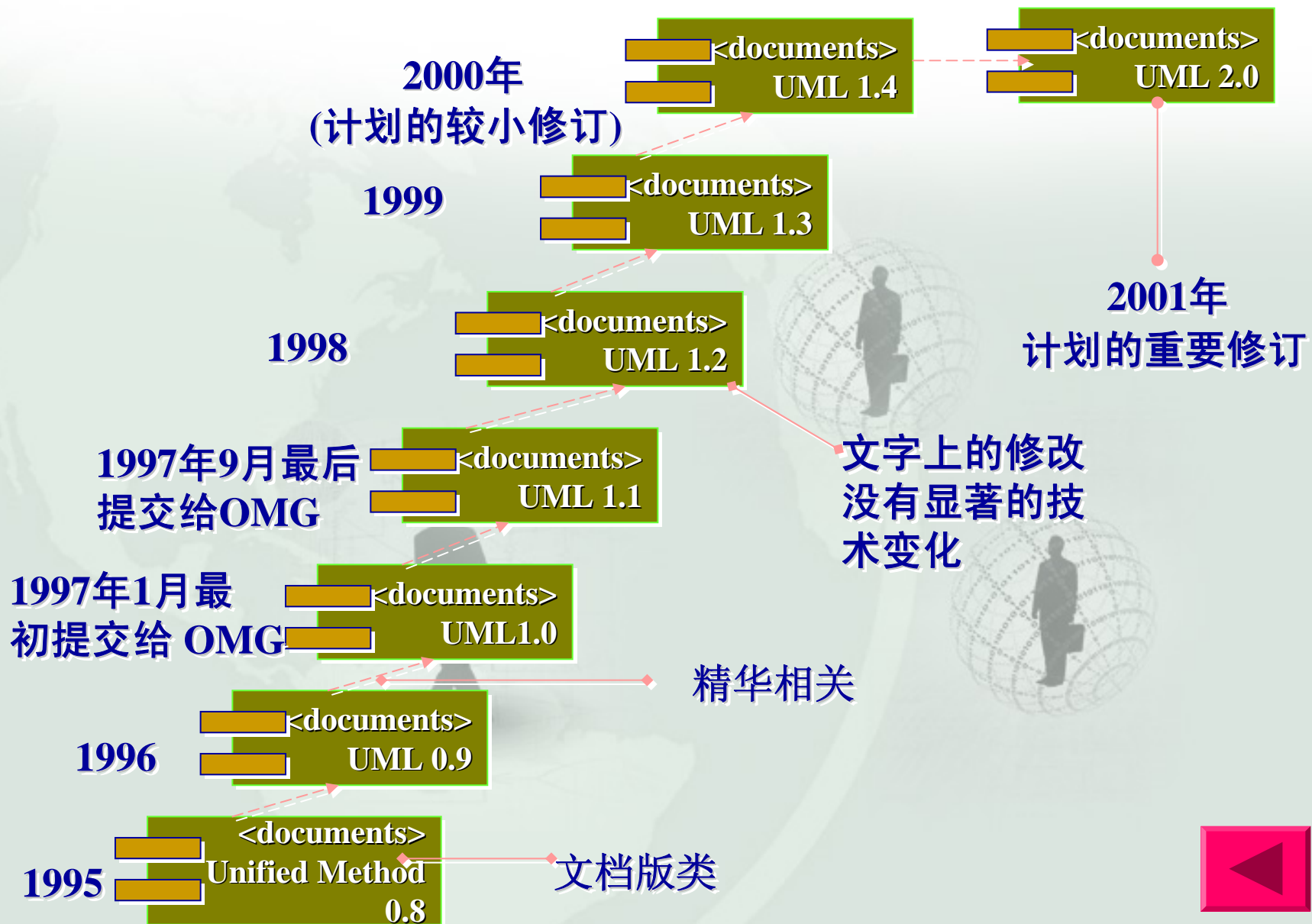
**1994年10月Jim Rumbaugh和Grady Booch共同合作把他们的OMT和Booch方法统一起来**

**1995年成为“统一方法” (Unified Method) 版本0.8**

**Ivar Jacobson加入,并采用他的用例(User case)思想,到1996年，成为“统一建模语言”版本0.9。**

**1997年1月UML版本1.0被提交给OMG组织，作为软件建模语言标准的候选**

**其后的半年多时间里，一些重要的软件开发商和系统集成商都成为“UML伙伴”，如IBM,Mircrosoft,HP等.1997年11月7日被正式采纳作为业界标准。**



### 面向对象方法的特点

- 以**对象**为基础，利用特定的软件模块，完成从对象客体的描述到软件结构之间的转换
- 避免了其它方法在开发过程中的**不一致性**和**复杂性**
- 系统的开发具有**简单性**、**统一性**
- 开发周期**短**，费用**低**



### 教学目的

- ◆通过讲解结构化方法、快速原型法和面向对象方法的区别，使学生更好地掌握常用的开发方法的特点
- ◆学会选择信息系统的开发方法

### 教学要求

- ◆ 了解常用开发方法以及这些方法存在的问题
- ◆ 根据现实问题，能够选择合适的开发方法
- ◆ 具备利用上述方法开发信息系统的能力

### 开发方法的选择

- ❑ 结构化方法与快速原型法的区别
- ❑ 结构化方法与面向对象方法的区别
- ❑ 原型化方法与结构化方法的结合
- ❑ 管理信息系统开发方法的选择





### 结构化方法与快速原型法的区别

#### 结构化方法

- 在系统建立之前**需求**就能被充分理解
- 它要求**严格划分开发阶段**，用规范的方法与图表工具有步骤地来完成各阶段的工作
- 每个阶段都**以规范的文档资料作为其成果**，最终得到满足用户需要的系统
- 结构化分析、结构化设计，结构化程序设计（简称**SA—SD—SP**方法）可用**瀑布模型来模拟**
- 各阶段的工作**自顶向下**从抽象到具体顺序进行
- 瀑布模型意味着在生命周期各阶段间存在着**严格的顺序**且相互依存。

### 结构化方法与快速原型法的区别

#### 优点

- 从系统整体出发，逻辑设计与物理设计分开
- 强调在整体优化的条件下“自顶而下”地分析和设计，保证了系统的整体性和目标的一致性
- 遵循用户至上原则
- 严格区分系统开发的阶段性
- 每一阶段的工作成果是下一阶段的依据，便于系统开发的管理和控制
- 文档规范化，按工程标准建立标准化的文档资料，便于后期的修改和维护。

### 结构化方法与快速原型法的区别

#### 缺点

- 用户或系统分析员和管理者之间的沟通困难
- 开发周期长，难于适应环境变化
- 结构化程度较低的系统，在开发初期难以锁定功能要求
- 开发过程复杂繁琐。

#### 适用范围

- 该方法适用于一些组织相对稳定、业务处理过程规范、需求明确，且在一定时期内不会发生大的变化的大型复杂系统的开发。

### 结构化方法与快速原型法的区别

#### 原型法

- 开发人员与用户共同确定系统的**基本要求和主要功能**
- **开发一个原型**
- 该原型是在较短时间内开发的一个实验性的、简单易用的小型系统
- 反复对原型进行**修改**，使之逐步完善，直到用户对系统完全满意为止
- 原型法的核心是用**交互的**、快速建立起来的原型**取代了**形式的、僵硬的（不易修改的）大型的**规格说明**
- 用户通过在计算机上实际运行和试用原型而向开发者提供真实的反馈意见
- 快速原型法的实现基础之一是可视化的第四代语言的出现



### 结构化方法与快速原型法的区别

#### 优点

- 符合人们认识事物的规律，系统开发循序渐进，反复修改，确保较好的用户满意度
- 开发周期短，费用相对少
- 由于有用户的直接参与，系统更加贴近实际
- 易学易用，减少用户的培训时间
- 应变能力强；降低开始风险和开发成本。

### 结构化方法与快速原型法的区别

#### 缺点

- 不适合开发大型的信息系统
- 开发过程管理要求高，整个开发过程要经过“修改—评价—再修改”的多次反复，开发难以控制
- 用户过早看到系统原型，误认为系统就是就是这个模样，易使用户失去信心
- 开发人员易将原型取代系统分析
- 缺乏规范化的文档资料，系统难于维护
- 如果用户合作不好，盲目纠错，会拖延开发进程



### 结构化方法与快速原型法的区别

#### 适用范围

- 处理过程明确，规模小，不太复杂的系统
- 用户需求不清，管理及业务不稳定，需求经常变化，涉及面窄的小型系统
- 不适合于大型、复杂的难以模拟的系统
- 存在大量运算、逻辑性强的处理系统
- 管理基础工作不完善、处理过程不规范
- 大量批处理的系统



### 面向对象方法与结构化方法的区别

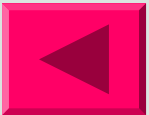
- 结构化方法强调过程抽象和模块化，将现实世界映射为数据流和加工
- 加工之间通过数据流进行通信
- 数据作为被动的实体，它是以过程（或操作）为中心来构造系统和设计程序的，数据模型和处理模型是分离的
- 在分析和设计阶段建立不同的模型，分析和设计存在不一致的问题
- 结构化方法采用“自顶向下”的开发方法
- 严格开发的阶段
- 按照功能划分设计的系统模块容易发生变化，使得开发出来的模块的可维护性欠佳
- 面向过程模式将数据与过程分离，若对某一数据结构做了修改，所有处理数据的过程都必须重新修订，这样就增加了很多的编程工作量

### 面向对象方法与结构化方法的区别

- 面向对象方法围绕现实世界的概念来组织模块，采用对象描述问题空间的实体
- 对象将数据和操作封装在一起，提供有限的接口
- 对象之间通过消息相互通信
- 它用程序代码模拟现实世界中的对象，使程序设计过程更自然、更直观
- 使用面向对象方法开发MIS时，工作重点在生命周期中的分析阶段
- 分析阶段得到的各种对象模型也适用于设计阶段和实现阶段
- 面向对象方法具有的继承性和封装性，支持软件复用，并易于扩充
- 面向对象方法是“自底向上”的开发过程，每个阶段的界限不是很严格

### 面向对象方法与结构化方法的区别

- 面向过程是以**功能**为中心来描述系统
- 面向对象是以**数据**为中心来描述系统，数据具有更强的稳定性
- 面向对象**模拟**了对象之间的**通信**
- 面向过程则通过**函数参数**和**全局变量**达到各过程模块**联系**的目的
- 面向对象把一个复杂的问题**分解**成多个能够完成独立功能的对象（类）
- 把这些对象组合起来去完成这个复杂的问题
- 采用面向对象模式就象在流水线上工作，我们最终只需将多个零部件（已设计好的对象）按照一定关系**组合**成一个完整的系统
- 这样使得软件开发更有效率。



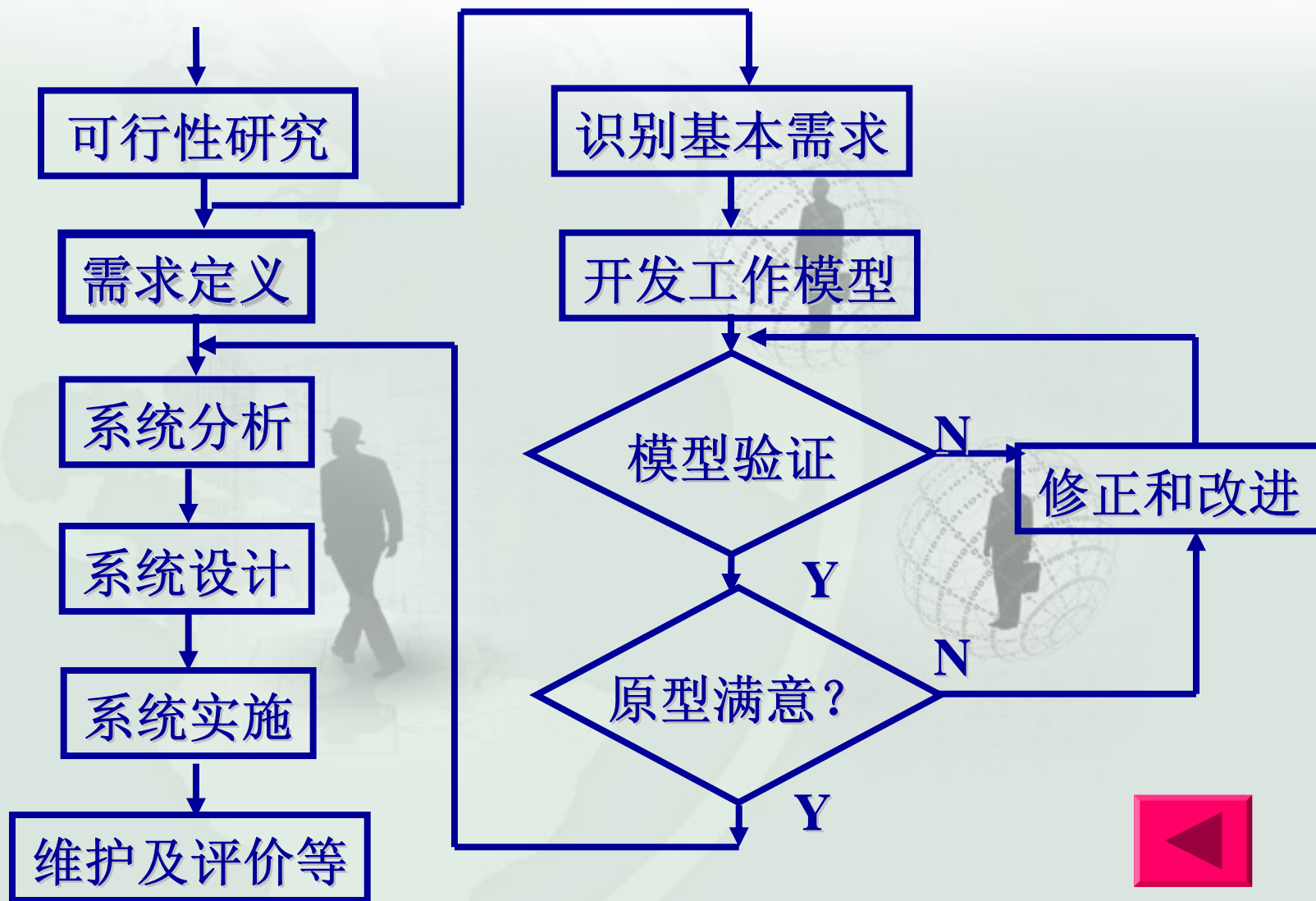


### 原型化方法与结构化方法的结合

- 原型的开发过程作为传统的SDLC方法的一个阶段，即需求定义阶段
- 研究型原型加强了SDLC的需求定义和功能分析阶段的分析



# 原型化方法与结构化方法的结合



### 系统开发方法的选择

系统开发方法涉及到以下因素：

(1) 应用的特点

需求和应用的类型、数据、过程、算法和环境等

(2) 方法的选择

各种方法的特点、应用的假设条件

(3) 资源分析

- 人力资源
- CASE工具的可利用性
- 时间、资金等方面的约束条件

### 系统开发方法的选择

- 要使应用与方法相匹配，并考虑到资源与约束等方面的情况
- 当企业应用领域的系统需求明确，稳定时，可采用结构化生命周期法
- 系统开发要求周期短，用户提不出明确、全面的需求，系统分析员对用户专业不熟悉并且很难定义用户需求时，可选用原型法开发系统
- 系统处在复杂多变的环境中，功能和数据类型庞大且复杂，不稳定且容易变化时，可采用面向对象的开发方法
- 从开发方法对使用者的要求看
  - 结构化生命周期法离计算机近一些
  - 原型法离用户近一些
  - 而面向对象方法则处于两者之间
- 在实际项目的开发中，常常是几种方法结合起来使用



### 小 结

- 管理信息系统的开发，根据不同的条件可以采用自而上而下、自下而上和两者结合的开发策略，选择自行开发、委托开发、联合开发和购买软件包等开发方式
- 目前在管理信息系统建设中还存在一些认识的偏差，开发问题，对管理信息系统开发的关键把握不准，教育、理论体系研究也落后
- 管理信息系统常用的开发方法有：结构化方法、快速原型法和面向对象的方法

### 小 结

- 结构化方法用于需求基本稳定的系统,开发严格按照阶段进行
- 原型法用于需求模糊或需求经常变化的系统,首先要了解用户的主要需求和基本功能,建立一个小的原型,根据用户使用的意见进行迭代
- 原型法难于进行控制
- 面向对象方法是一种自底向上的方法,稳定性和扩充性好
- 三种方法各有优缺点,需要根据应用系统的特点,方法本身的特点和资源情况进行考虑

