

# 软件工程

# 学习“软件工程”的目的和意义

- 学会如何在现代IT企业的环境中做一个成功者；
- 学会如何做世界级的、高质量的研究；
- 学会如何创建大规模的软件产品。

——百度总裁，原微软全球资深  
副总裁、微软亚洲研究院院长、  
首席科学家 张亚勤



# The Challenges and Opportunities of the Global Software Industry.

## 一、全球软件产业的现状、趋势与挑战

1. 软件产业的网络化趋势
2. 软件产业的服务化趋势
3. 软件产业的全球化趋势

### 智能网络 (Intelligent Web)

- 从静态网到动态网
- 从被动方式到主动方式
- 从呈现信息和浏览的窗口到智能生成的平台
- 从HTML到XML



# The Challenges and Opportunities of the Global Software Industry.

## 一、全球软件产业的现状、趋势与挑战

1. 软件产业的网络化趋势

2. 软件产业的服务化趋势

3. 软件产业的全球化趋势

相对目前“打包式”软件，为用户提供智能化的“服务式”软件。



# The Challenges and Opportunities of the Global Software Industry.

## 一、全球软件产业的现状、趋势与挑战

1. 软件产业的网络化趋势

2. 软件产业的服务化趋势

3. 软件产业的全球化趋势

- 强调开放性、交互性
- 支持主流的开放式标准
- 人才的全球化

对中国软件企业有什么启示和挑战？



## 二、现代软件开发对人才的要求

Talent Beyond Technology \_What Kind of Talent  
We Need for Modern Software Development

- 扎实的基础
- 创新、独立的工作能力
- 主人翁精神和团队精神
- 沟通与协调能力
- 成就感强、有激情
- 自觉地干好工作
- 契而不舍，从错误中学习



## 在微软流传着这样一个故事

有一次比尔·盖茨和上帝谈话，上帝对他说：“地球明天就要毁灭了！由于你是如此的成功，因此我特许你从地球上带一样东西到天堂！你想带什么？”

比尔·盖茨回答：“请允许我从微软选300个最优秀的人，我要把他们带到天堂！”

尽管微软的windows, Exchang, SQL, Office等都非常重要，但如果没有象你们这些人才，微软也将一无所有。你们对我们公司和微软来说是处于第一位的。



### 三、信息化建设中复合型软件人才的需求

我国复合型软件人才匮乏，培养适应迅速变化的市场需求的复合型软件人才已是迫在眉睫。

复合型软件人才一指具备不同专业背景，并具有市场观念的开发管理、工程管理和软件经营等技术和经验的软件人才。





## 四、软件开发中的错误观点

1. 只要掌握了1-2门程序设计语言，进行软件开发就没有问题。
2. 软件需求分析很困难，不管三七二十一先把软件做了再说，反正软件是灵活的，随时可以修改。
3. 只要有最好的开发工具、最好的计算机，一定能做出优秀的软件。

总之，错误认为：软件就是程序，开发软件就是编写程序。



# 本课程学习的主要内容

1. 传统的软件工程方法
2. 面向对象的建模技术——UML统一建模语言
3. 软件项目管理及CMM
4. 软件工程课程设计

本课程以传统的软件工程技术为基础，主要讨论如何应用软件工程的最新技术——UML统一建模语言进行软件开发，提高软件开发的水平和技术。



# 课程特点与安排

## 1. 课程特点:

实践性强、发展迅速

## 2. 课程安排(32学时):

课堂教学 (24学时) — 授课、案例分析、讨论

上机实验 (8学时, 课内外结合完成, 实验报告)



## 教材及主要参考书

### 1. 软件工程——理论与实践（第3版）

作者：许家珩, 白忠建, 吴磊编著

出版社：高等教育出版社

### 2. 软件工程——方法与实践（第2版）

作者：许家珩 主编, 白忠建, 吴磊编著

出版社：电子工业出版社



# 教材及主要参考书

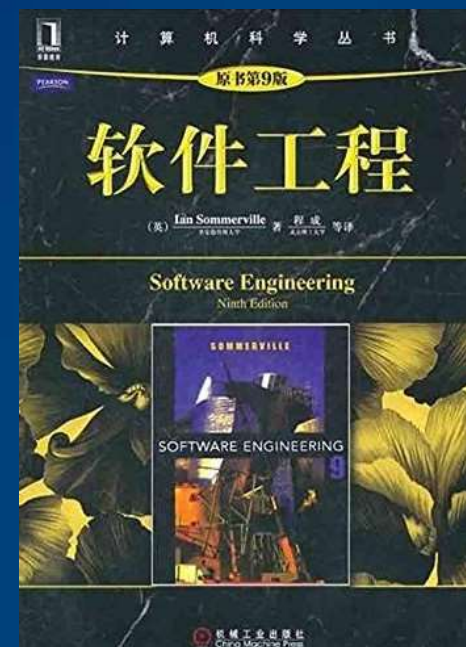
## 3. 软件工程

Software Engineering, 9th Edition

作者：（英）Ian Sommerville

出版社：机械工业出版社（影印版）

本书是英国著名软件工程学家 Ian Sommerville 系统介绍软件工程理论的力作，以要求极高的一类系统为实例，精辟透彻地阐述了软件工程的内涵。



## 参考网站

<http://www.sei.cmu.edu>

卡内基·梅隆大学软件工程研究所.

<https://www.ibm.com/software/rational>

IBM Rational 软件开发平台.

<http://www.rational.com>. Rational公司.

<http://requirements-engineering.org/>

软件需求工程国际会议网站



1

# 第一章 概述



# 1.1 软件工程的产生和发展



软件工程 (Software Engineering) 是在克服 60 年代末所出现的“软件危机”的过程中逐渐形成与发展的。

随着互联网平台和环境的迅速发展，对软件工程的发展产生了变革性的巨大影响，网格计算、普适计算、中间件、网构软件等一批新技术和理念，从网络资源共享与管理，人机交互及软件服务等各方面提出了软件技术的新课题。





40多年来，软件工程已经历了四个重要发展阶段：

1. 第一代软件工程 — 传统的软件工程
2. 第二代软件工程 — 对象工程
3. 第三代软件工程 — 过程工程
4. 第四代软件工程 — 构件工程



40多年来，软件工程已经历了四个重要发展阶段：

1. 第一代软件工程 — 传统的软件工程

2. 第二代软件工程 —

3. 第三代软件工程 —

4. 第四代软件工程 —

60年代末到70年代为了克服“软件危机” (Software crisis) 提出“软件工程”的名词，将软件开发纳入工程化的轨道，基本形成软件工程的概念、框架、技术和方法。称为传统的软件工程。



40多年来，软件工程已经历了四个重要发展阶段：

1. 第一代软件工程 — 传统的软件工程

2. 第二代软件工程 — 对象工程

3. 第三代软件工程

4. 第四代软件工程

80年代中到90年代，面向对象的方法与技术得到发展，研究的重点转移到面向对象的分析与设计，演化成为一种完整的软件开发方法和系统的技术体系，称为对象工程。



40多年来，软件工

1. 第一代软件工程

2. 第二代软件工程 — 软件工程

3. 第三代软件工程 — 过程工程

4. 第四代软件工程 — 构件工程

80年代中开始，人们在软件开发的实践过程中认识到：提高软件生产率，保证软件质量的关键是“软件过程”，是软件开发和维护中的管理和支持能力，逐步形成软件过程工程。



- 40多年来，软件的开发方法取得重要进展，软件系统的开发可通过使用现成的可复用构件组装完成，而无需从头开始构造，以此达到提高效率和质量，降低成本的目的。称为构件工程。
- 90起年代，基于构件（Component）
1. 第一代软件工程
  2. 第二代软件工程
  3. 第三代软件工程 — 过程工程
  4. 第四代软件工程 — 构件工程



软件工程是一门新兴的边缘学科,涉及的学科多,研究的范围广,研究的主要内容有以下几方面:

- 软件开发方法、技术
- 软件开发工具及环境
- 软件管理技术
- 软件规范 (国际规范)

} 软件开发技术

} 软件管理技术



## 1.2 软件工程过程

为了克服软件危机，人们从其他产业的工业化生产得到启示，于是在68年北大西洋公约的软件可靠性会议（NATO）上，首次提出了“软件工程”的概念。提出了在软件生产中采用工程化的方法，采用一系列科学的、现代化的方法技术来开发软件。这种工程化的思想贯穿到软件开发和维护的全过程。





● IBM公司的 OS/360，共约10年；经费达数亿美圆，而结果却达2000个以上，系统根本无法正常运行。项目负责人Brooks这样描述开发过程的困难和混乱：“...在泥潭中作垂死挣扎，挣扎得越猛，泥浆就沾得越多...没有一个野兽能够逃脱淹没在泥潭中的命运...”

● 1963年美国飞往火星的火箭爆炸，造成1000万美元的损失。原因是FORTRAN程序：

**DO 5 I=1, 3**

误写为：**DO 5 I=1 . 3**

● 1967年苏联“联盟一号”载人宇宙飞船在返航时，由于软件忽略一个小数点，导致在进入大气层时因打不开降落伞而烧毁。



“软件危机” (Software crisis) 的出现是由于软件的规模越来越大，复杂度不断增加，软件需求量增大。而软件开发过程是一种高密集度的脑力劳动，软件开发的模式及技术不能适应软件发展的需要。致使大量质量低劣的软件涌向市场，有的花费大量人力财力，而在开发过程中就夭折。

“软件危机”主要表现在两个方面：

- (1) 软件产品质量低劣，甚至开发过程就夭折。
- (2) 软件生产率低，不能满足需要。

# 软件工程过程

## (Software engineering process)

是指在软件工具的支持下，所进行的一系列软件开发和进化的活动。

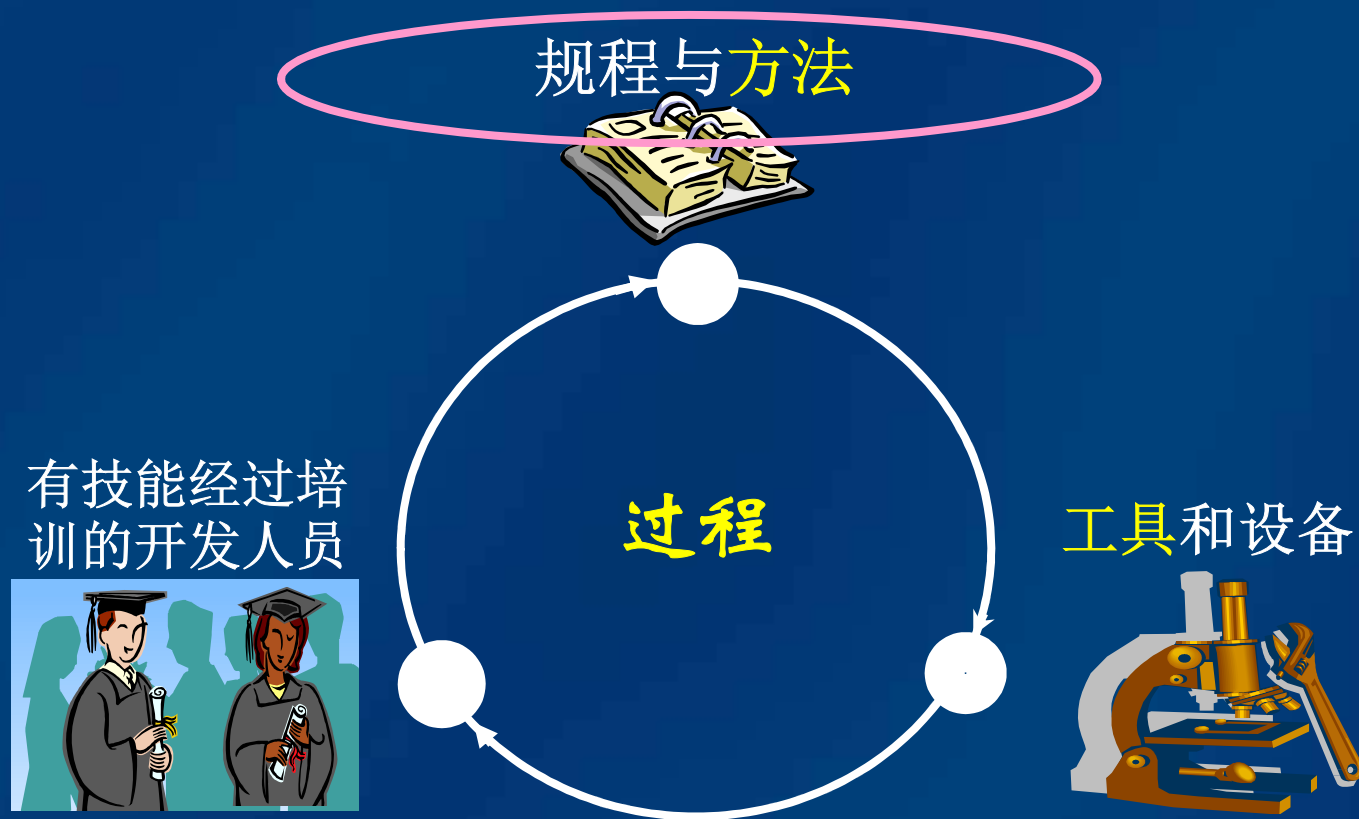
通常包括以下四类基本过程：

- 1、软件规格说明：规定软件的功能及其运行环境。
- 2、软件开发：产生满足规格说明的软件。
- 3、软件确认：确认软件能够完成客户提出的要求。
- 4、软件演进：为满足客户的变更要求，软件必须在使用进的过程中演进。



# 软件工程过程

(Software engineering process)



# 软件生命周期 (SDLD) —瀑布模型

软件包括程序及软件开发过程所产生的所有文档。



软件生命期模型



## 1.3 软件过程模型

软件过程模型是描述软件开发过程中各种活动如何执行的模型。因此又称为**软件开发模型**。

软件过程模型是对软件开发实际过程的抽象和简化。

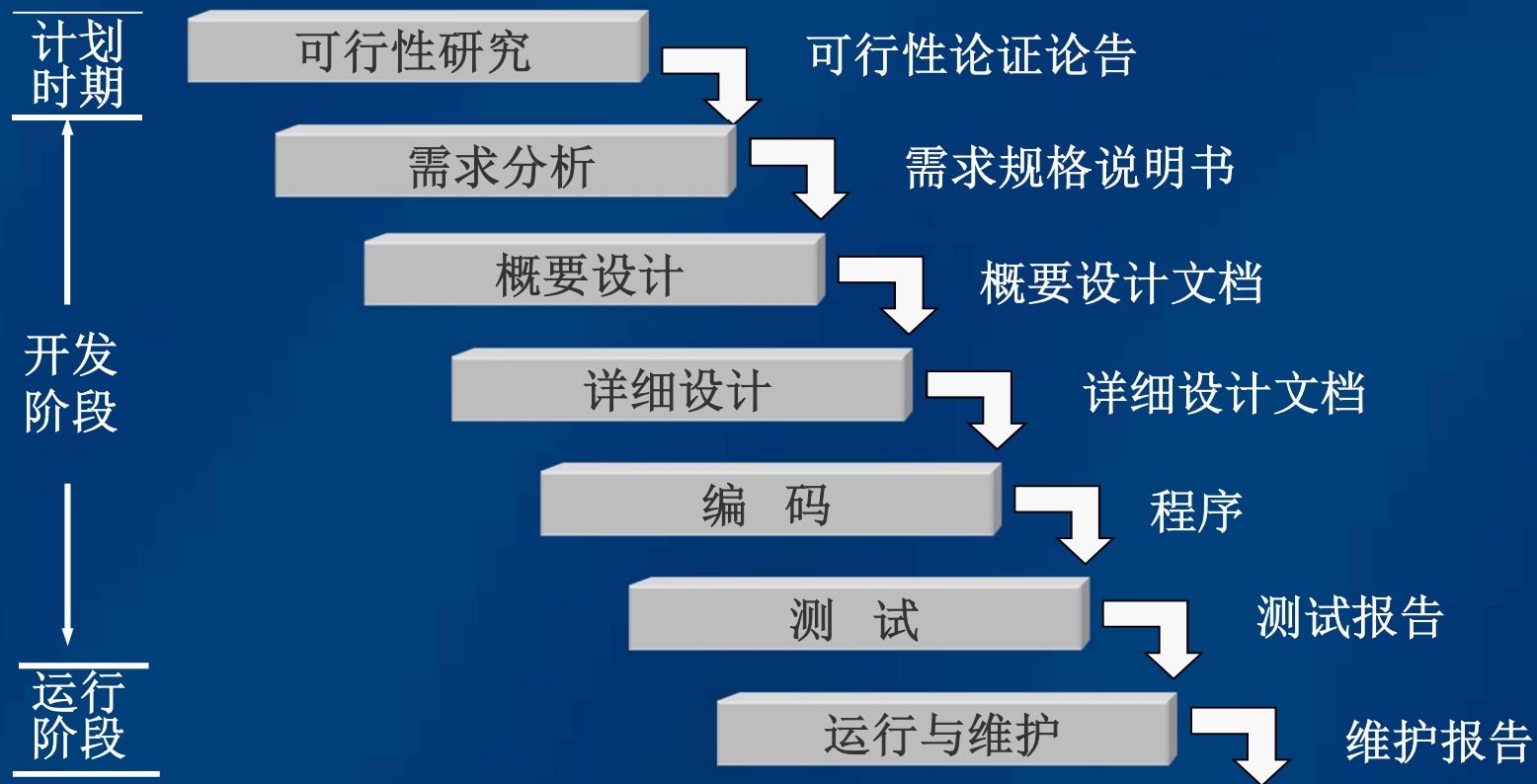
目前典型的软件开发模型有：

瀑布模型、增量模型、螺旋模型、喷泉模型、变换模型和基于知识的模型等。

不同的开发方法有不同的软件过程模型。



# 1. 瀑布模型



瀑布模型在软件开发的前期起到重要作用，但逐渐暴露出其缺陷，即将充满回溯的软件开发过程硬性分割为几个阶段。



## 2. 增量模型 (incremental model)

增量模型是一种非整体开发的模型。是一种进化式的开发过程。

根据增量的方式和形式的不同，分为：

- 基于瀑布模型的渐增模型
- 基于原型的快速原型模型

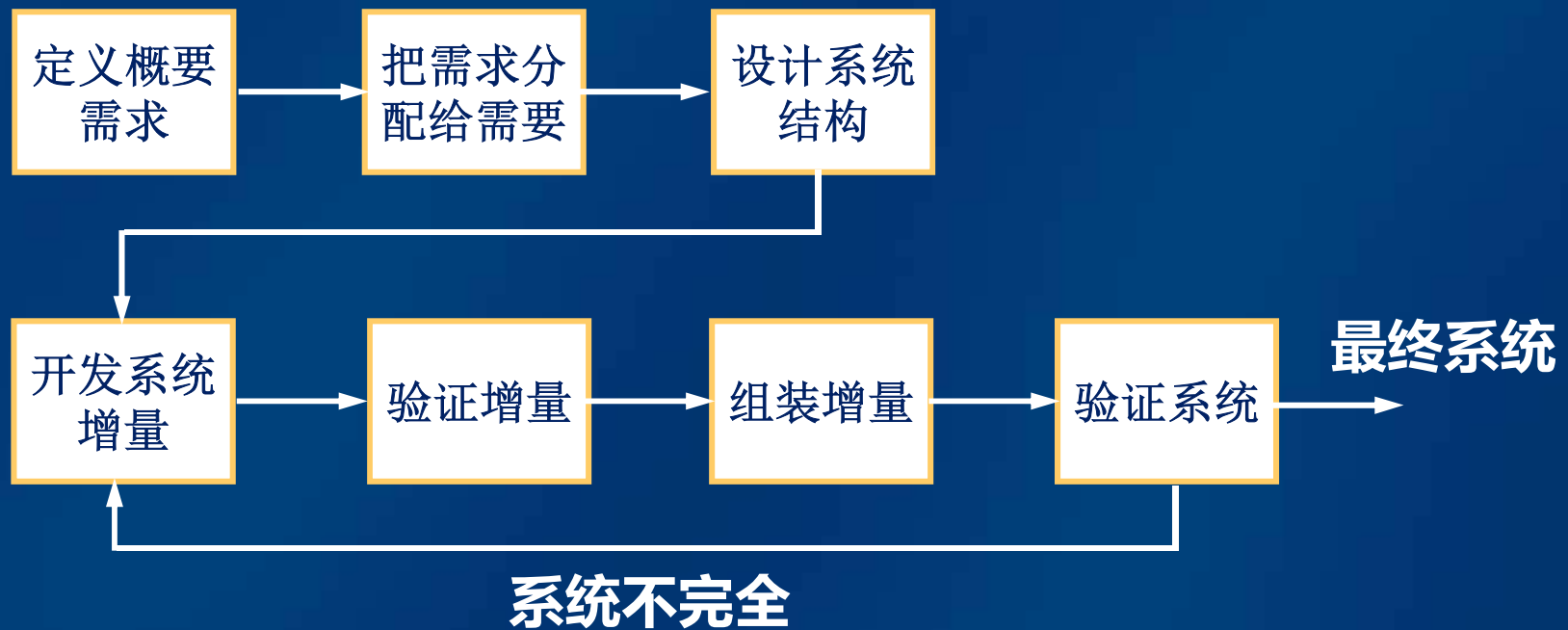
该模型具有较大的灵活性，适合于软件需求不明确、设计方案有一定风险的软件项目。

增量模型和瀑布模型之间的本质区别是什么？



## 2. 增量模型

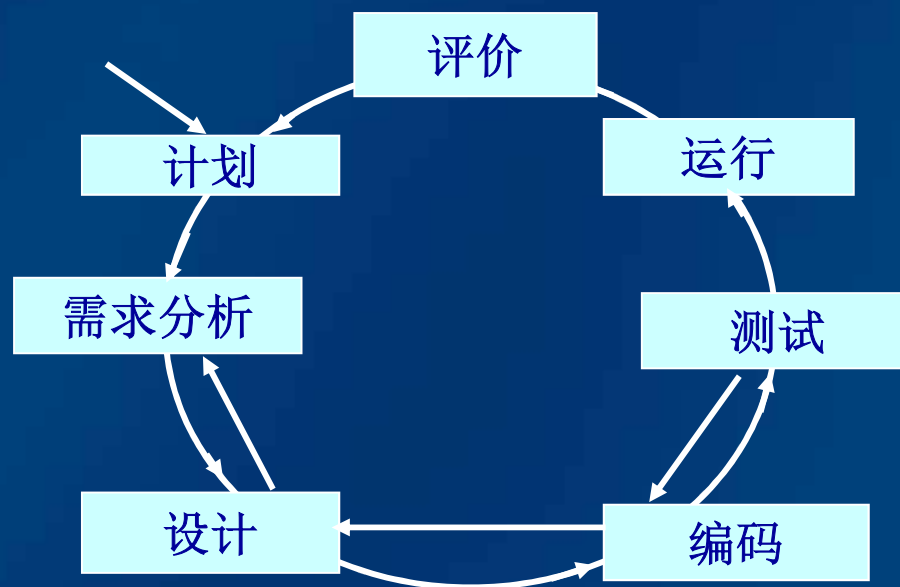
下图是对增量模型的一般描述。





### 3. 循环模型

为了对瀑布模型进行改进，描述软件开发过程中可能的回溯，采用循环模型。



循环模型



## 4. 速成原型模型

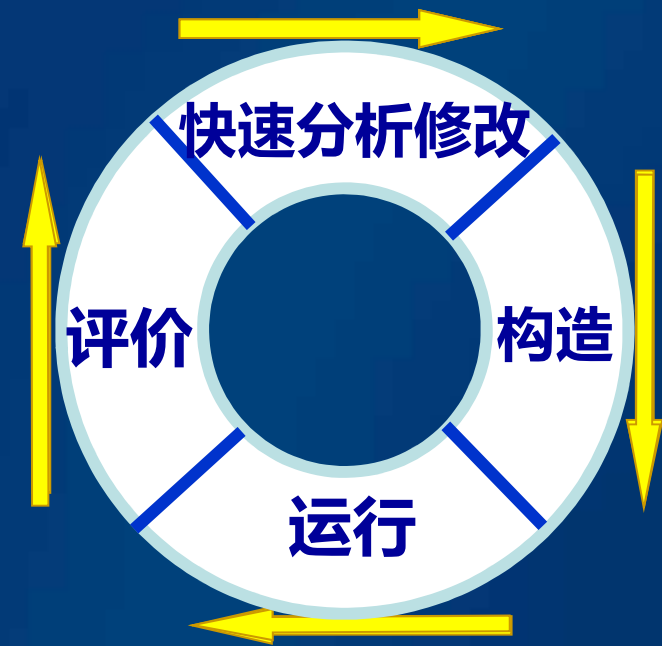
速成原型的工作模型是一个循环的模型。

1. **快速分析** 快速确定软件系统的基本要求，确定原型所要体现的特征（界面、总体结构、功能、性能）

2. **构造原型** 考虑主要特征，快速构造一个可运行的系统。有三类原型：用户界面原型、功能原型、性能原型。

3. **运行和评价原型**

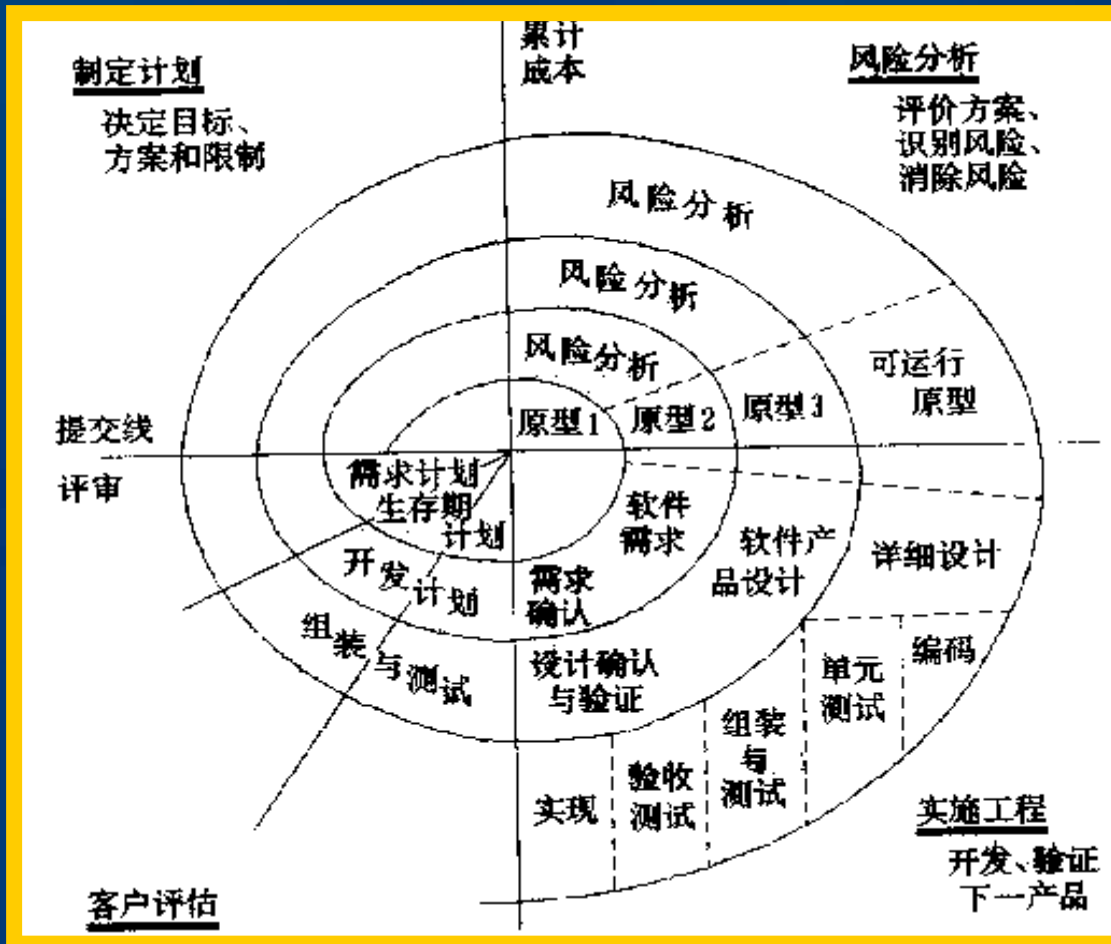
4. **修改与改进**



原型化工作模型

## 5. 螺旋模型

对大型软件,需要多个原型描述系统的生存期,螺旋模型将瀑布模型与原型化模型结合起来,并加入了风险分析。



螺旋模型将开发过程分为几个螺旋周期,每个螺旋周期可分为4个工作步骤:

第一,确定目标、方案和限制条件;

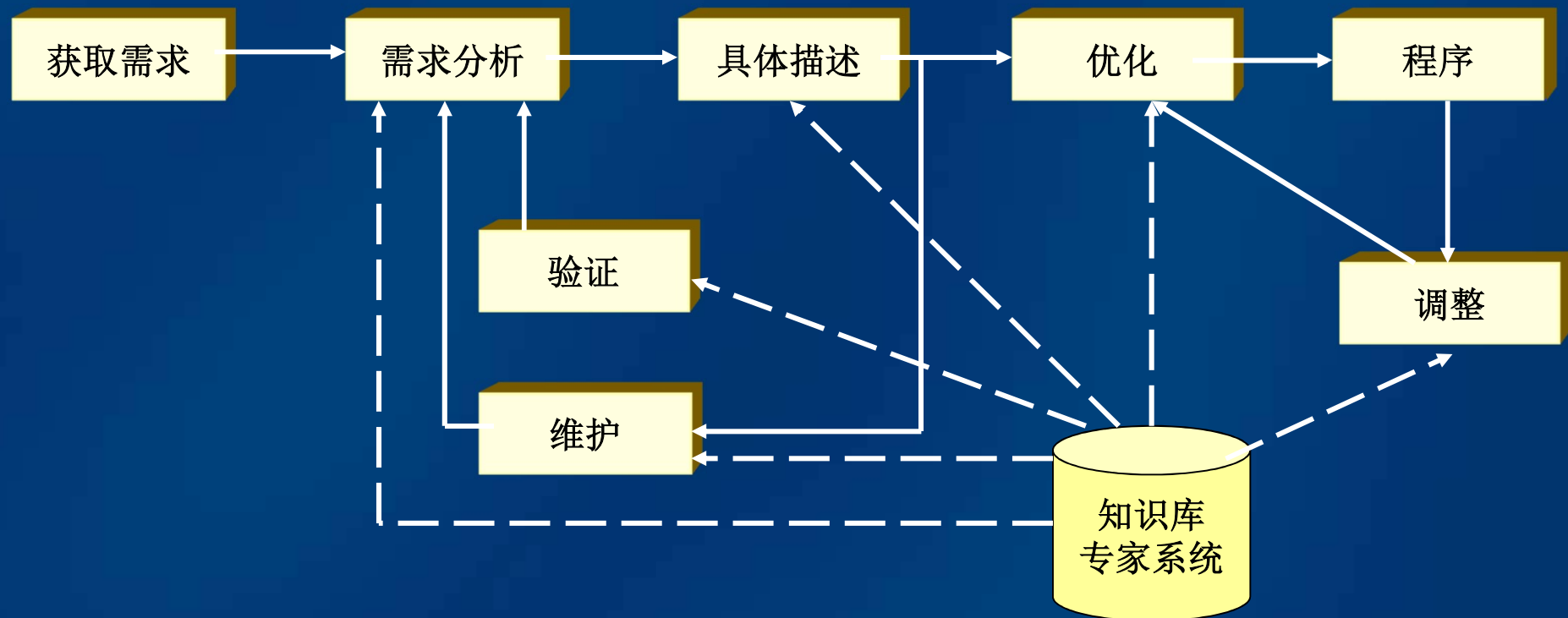
第二,评估方案、标识风险和解决风险;

第三,开发确认产品;

第四,计划下一周期工作。

## 6. 智能模型(intelligent model)

也称为基于知识的软件开发模型，是知识工程与软件工程相结合的软件开发模型。



智能模型

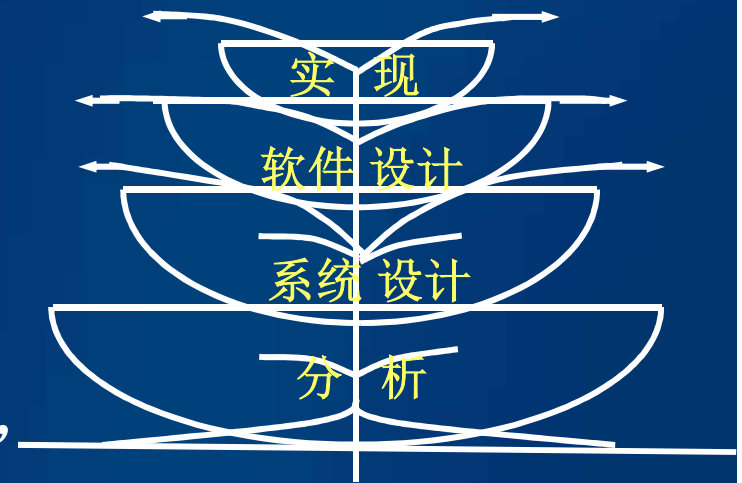


## 7. 喷泉模型

该模型是由B. H. Soilers和J. M. Edwards于1990年提出。它克服了瀑布模型不支持软件重用和多项开发活动集成的局限性，喷泉模型使开发过程具有迭代性和无间隙性。适宜面向对象的方法。

其特点如下：

1. 开发过程有分析、系统设计、软件设计和实现4个阶段。
2. 各阶段相互重叠，它反映了软件过程并行性的特点。
3. 以分析为基础，资源消耗成塔型。
4. 反映了软件过程迭代性的自然特性，从高层返回低层无资源消耗。
5. 强调增量开发，整个过程是一个迭代的逐步提炼的过程。



喷泉模型

## 1.4 软件开发方法

软件开发的目标是要在规定的投资和时间内，开发出符合用户的需求，高质量的软件，为此需要有成功的开发方法。

软件开发方法可分为：

- 面向过程的开发方法
- 面向对象的开发方法
- 基于构件的开发方法





## 1.4.1 结构化开发方法

### 结构化开发方法 (Structured Developing Method)

是现有的软件开发方法中最成熟，应用最广泛的方法，主要特点是快速、自然和方便。

结构化方法总的指导思想自顶向下、逐步求精。它的基本原则是功能的分解与抽象。

#### 一、结构化开发方法的组成

结构化程序设计方法 SP法 (Structured Program)

结构化设计方法 SD法 (Structured Design)

结构化分析方法 SA法 (Structured Analysis)

SA, SD, SP 法相互衔接，形成了一整套开发方法。



## 1.4.2 原型化开发方法

### 什么是原型化方法 (Prototyping Method)

**原型**——是软件的一个早期可运行的版本，它反映了最终系统的部分重要特性。

原型化方法的基本思想是花费少量代价建立一个可运行的系统，使用户及早获得学习的机会。

原型化方法又称速成原型法 (Rapid Prototyping)。





## 1.4.2 原型化开发方法

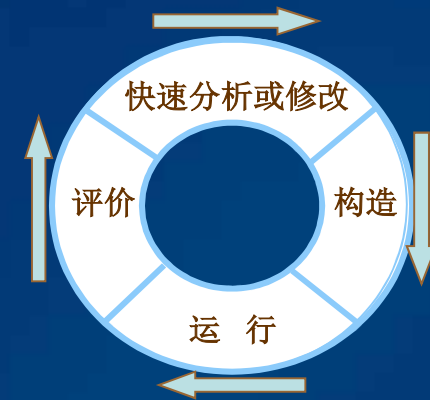
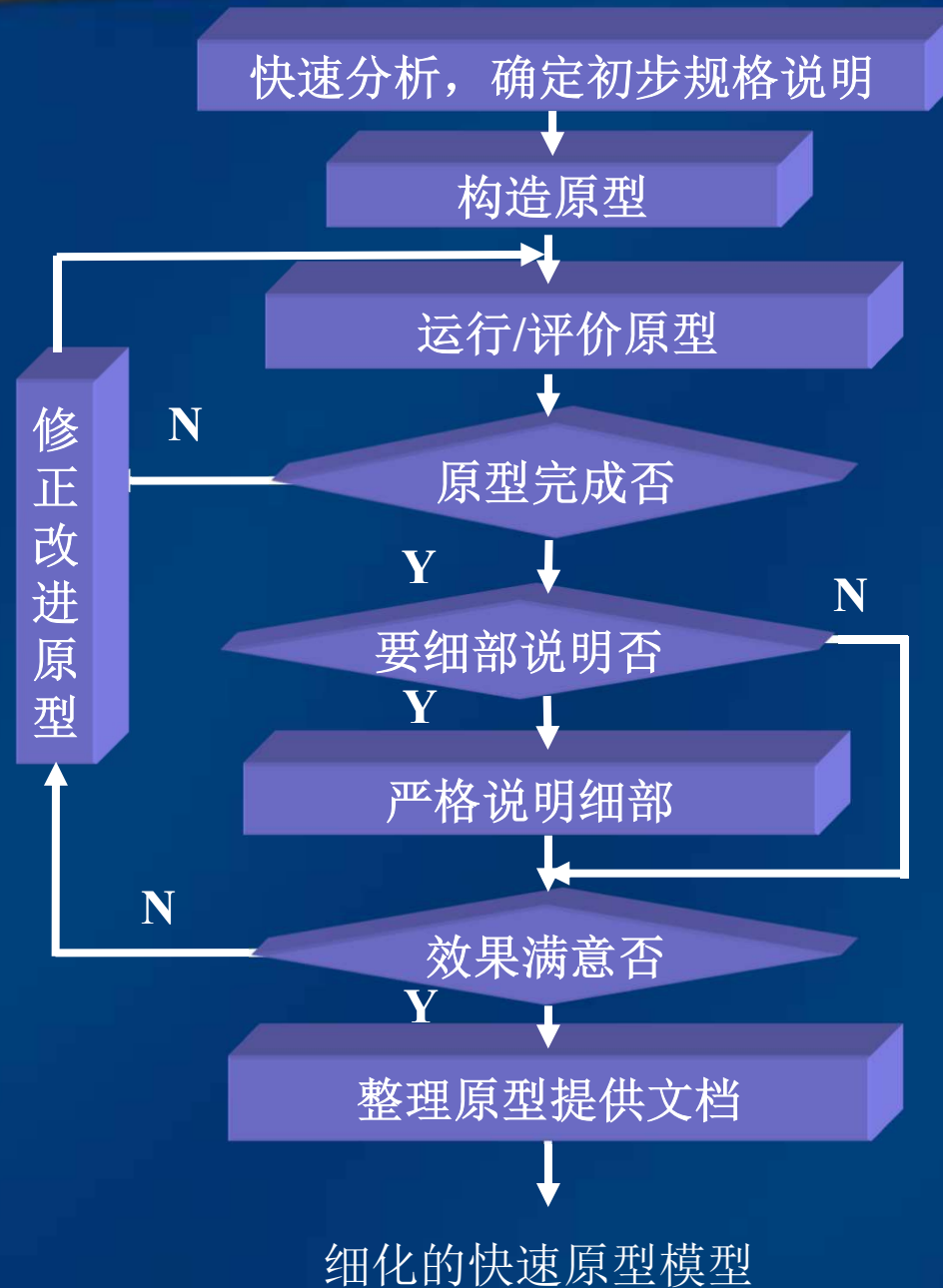
### 1、废弃 (throw away) 型

**RSP法** (Rapid Specific Prototyping) 快速建立需求规格原型。

### 2、追加 (add on) 型

**RCP法** (Rapid Cyclic Prototyping) 快速建立渐进原型法。采用循环渐进的开发方式，对系统模型作连续精化。





### 1.4.3 面向对象的开发方法

**面向对象方法**（OOSD(Object-Oriented Software Development)）是80年代推出的一种全新的软件开发方法。非常实用而强有力，被誉为90年代软件的核心技术之一。

**其基本思想是：**对问题领域进行自然的分割，以更接近人类通常思维的方式建立问题领域的模型，以便对客观的信息实体进行结构和行为的模拟，从而使设计的软件更直接地表现问题的求解过程。面向对象的开发方法以对象作为最基本的元素，是分析和解决问题的核心。



# 什么是面向对象

Coad和Yourdon给出一个面向对象的定义：

面向对象 = 对象 + 类 + 继承 + 消息

面向对象的开发方法,是以对象作为最基本的元素,来构造系统的,因此对象是分析和解决问题的核心。



# 面向对象的基本概念

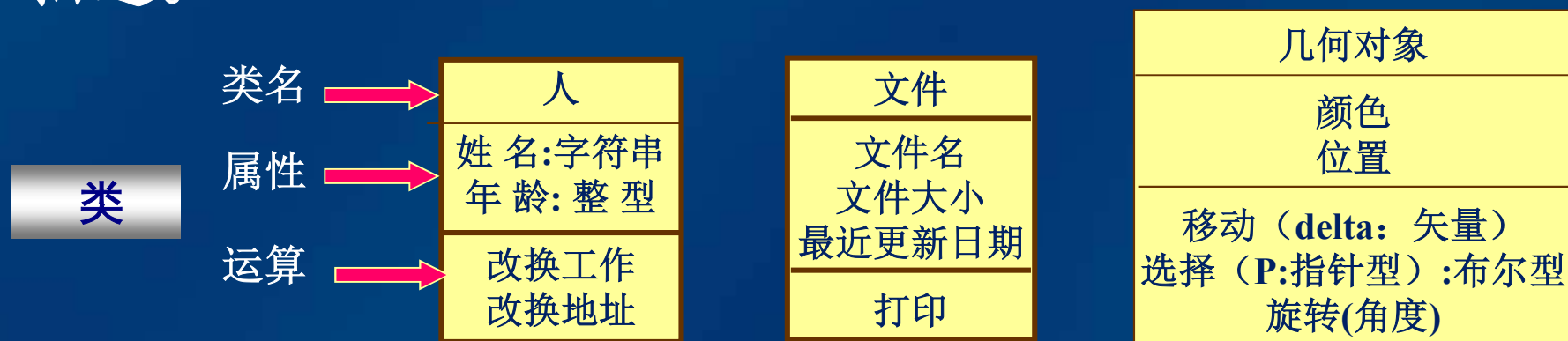
理解面向对象的基本概念对于学习和掌握面向对象的开发方法是十分重要的。

- 对象(Object)
- 类(Class)
- 继承(Inheritance)
- 消息(Information)
- 多态性(Polymorphism)

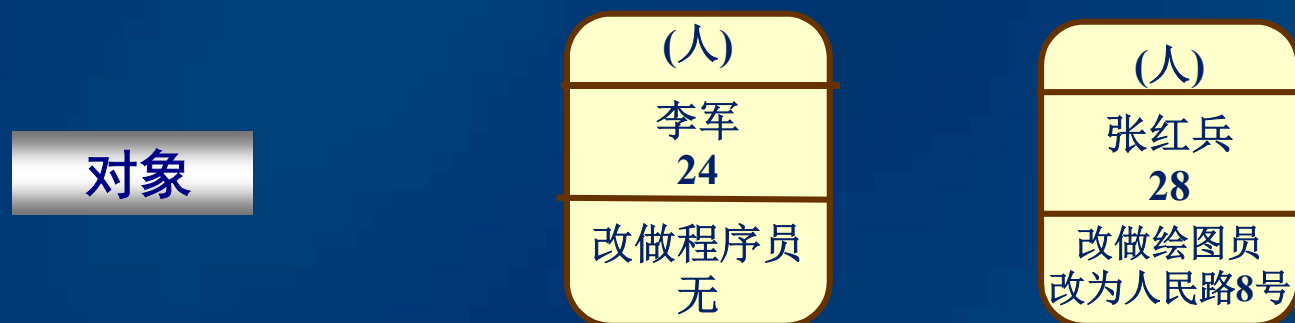


# 1、对象和类

类和对象一般采用“名字”、“属性”和“运算”来描述。



对象类的描述



对象的描述



## 2、继承 (*Inheritance*)

继承是使用现存的定义作为基础，建立新定义的技术。是父类和子类之间共享数据结构和方法的机制，这是类之间的一种关系。

继承性分：

单重继承：一个子类只有一个父类。即子类只继承一个父类的数据结构和方法。

多重继承：一个子类可有多个父类。继承多个父类的数据结构和方法。





### 3、消息 (Message)

消息就是向对象发出的服务请求（互相联系、协同工作等）。对象之间的联系可表示为对象间的消息传递，即对象间的通讯机制。

一个消息应该包含以下信息：消息名、接收消息对象的标识、服务标识、消息和方法、输入信息、回答信息。

在对象的操作中当一个消息发送给某个对象时，消息包含接收对象去执行某种操作的消息。



## 4、多态性

多态性 (Polymorphism) 是指相同的操作或函数、过程作用于不同的对象上并获得不同的结果。

即相同的操作的消息发送给不同的对象时，每个对象将根据自己所属类中定义的操作去执行，产生不同的结果。

例如：“绘图”操作，作用在“椭圆” 和“矩形” 上，画出不同的图形。



## 多态性的实现

实现多态性的基本步骤(以VC为例):

- (1) 在基类中, 定义成员函数为虚函数(virtual);
- (2) 定义基类的公有(public)派生类;
- (3) 在基类的公有派生类中“重载”该虚函数;
- (4) 定义指向基类的指针变量, 它指向基类的公有派生类的对象。

**注意:** 重载虚函数不是一般的重载函数, 它要求函数名、返回类型、参数个数、参数类型和顺序完全相同。



# 面向对象开发方法的组成

OOD由三部分组成：

OOA (Object-Oriented Analysis) 面向对象的分析

OOD (Object-Oriented Design) 面向对象的设计

OOP (Object-Oriented Program) 面向对象的程序设计



强调的是对一个系统中的对象特征和行为的定义。建立系统的三类模型。



与OOA密切配合顺序实现对现实世界的进一步建模。



是面向对象的技术中发展最快的，使用面向对象的程序设计语言，进行编码。



面向对象的方法的代表性成果有：

1. B. Henderson-sellers 和 J. m. Edwards提出的面向对象软件生存期喷泉模型及面向对象的系统开发方法。
2. G. Booch提出的面向对象的开发方法。
3. P. Coad和E. Yourd提出的OOA和OOD法。
4. J. Rumbaugh等人提出的对象建模技术(OMT)。
5. Jacobson 提出的面向对象的软件工程(OOSE)。
6. G. booch J. Rumbough Jacobson等人在Booch方法、OMT和OOSE的基础上推出了统一建模语言UML。



## 1.4.4. 敏捷软件的开发

敏捷软件开发是一种以人为核心、迭代式的开发方法，是为解决软件项目规模和复杂度的增加，以及需求常发生变化而影响软件按时交付的矛盾运用而生的。

核心价值观：

- |           |    |         |
|-----------|----|---------|
| ● 人和交互    | 胜过 | 过程和工具   |
| ● 可运行的软件  | 胜过 | 面面俱到的文档 |
| ● 与客户协作   | 胜过 | 合同谈判    |
| ● 对变更及时处理 | 胜过 | 遵循计划    |

敏捷开发更强调与客户的协作、人之间的交互与团队的协作。





## 1.4.4. 敏捷软件的开发

按照敏捷软件开发的思路推出的许多实践方法中，最具代表性的是XP法。

### XP方法简介

又称极限编程(Extreme Programming, XP)。  
由kent Beck于1999年提出来的。

以用户需求作为软件开发的最终目标，是一种以实践为基础的软件工程过程。

极限编程强调测试，强调代码质量和及早发现问题，以适应环境和需求的变化。





# XP方法简介

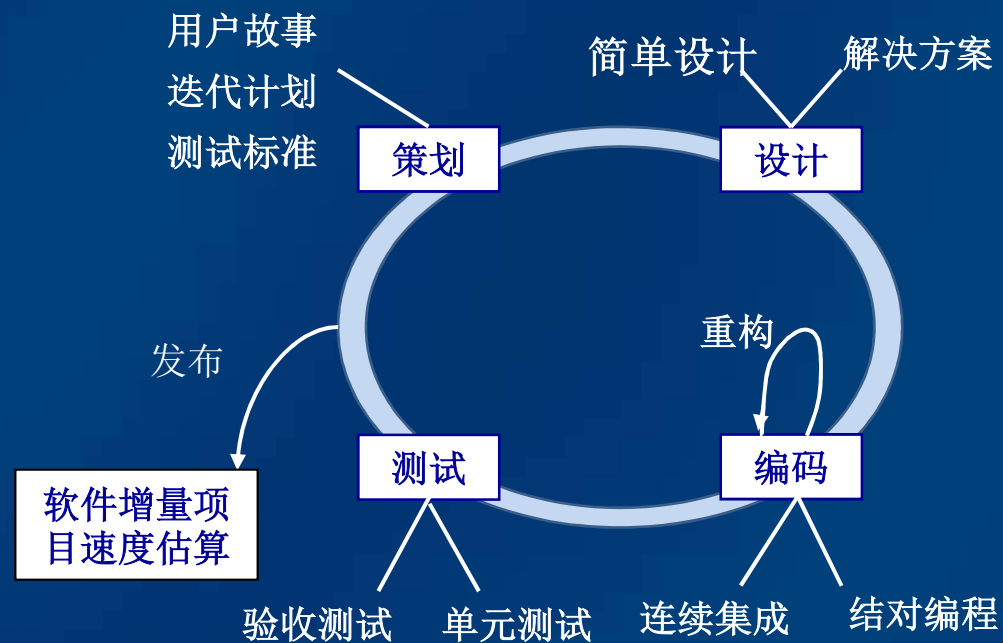
## 核心价值观

- **沟通(Communication)**：指开发人员与用户、开发人员间频繁而有效的信息交流，充分理解用户需求，是项目成功的关键。
- **简单(Simplicity)**：在满足用户需求的前提下，软件开发全过程及过程中的产品都应该尽量简单。
- **反馈(Feedback)**：及时、准确的信息反馈，能及时发现开发工作中的问题和偏差并纠正。
- **勇气(Courage)**：采用敏捷开发方法，是一种挑战，是需要勇气和信心；需要团队密切协作，只有十分需要的文档才写，即使写也要简单明了，这也需要勇气。

# XP方法简介

在其核心价值观的指导下，XP方法提出规划策略、小型发布等十二项最佳实践。

**XP方法的开发过程：**使用面向对象方法作为开发范型。包含策划、设计、编码和测试4个框架活动的规则和实践。



## 1.5 软件工具与集成化环境

### 一、软件工具

软件工具是用来辅助软件的开发、运行、维护、管理和支持等活动的软件系统。

按软件工具发展过程分：

- 1、孤立的单个软件开发工具；
- 2、集成化的CASE环境。

也称为软件开发环境(Software Development Environme)  
或软件工程环境(Software Engineering Environment)

是包括方法、工具和管理等多种技术在内的综合系统，应具备①紧密性 ②坚定性 ③可适应性 ④可移植性等特点。



## 1.5 软件工具与集成化环境

### 一、软件工具

按照软件工程过程分：

#### 1、软件开发工具

分析工具、设计工具、编码工具和调试工具等

#### 2、软件维护工具

版本控制工具、文档管理工具、开发信息库工具、  
逆向工程工具、逆向工程工具

#### 3、软件管理与支持工具

项目管理工具、配置管理工具、软件评价工具和风  
险分析工具

