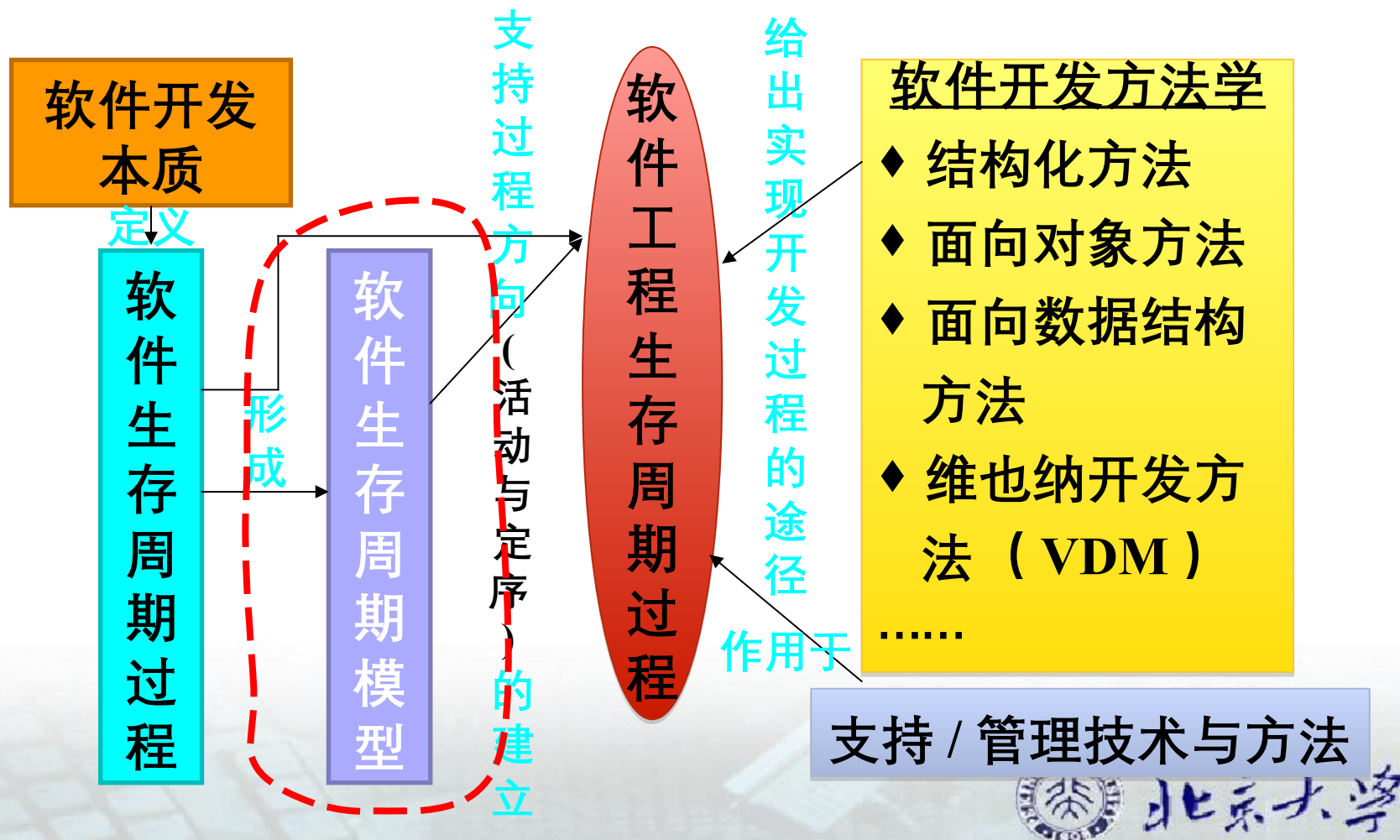


# ---- 软件生存周期模型

## 2、 开发活动的组织框架

### -- 软件生存周期模型





## 1) 基本概念

软件生存周期模型 **IEEE Standard 12207.0-1996**

把一个软件生存周期模型描述为：一个包括软件产品开发、运行和维护中有关过程、活动和任务的框架，覆盖了从该系统的需求定义到系统的使用终止。

**中国计算机科学与技术百科全书**

称软件生存周期模型为“软件开发模型”，并把它定义为：软件过程、活动、任务的结构框架。



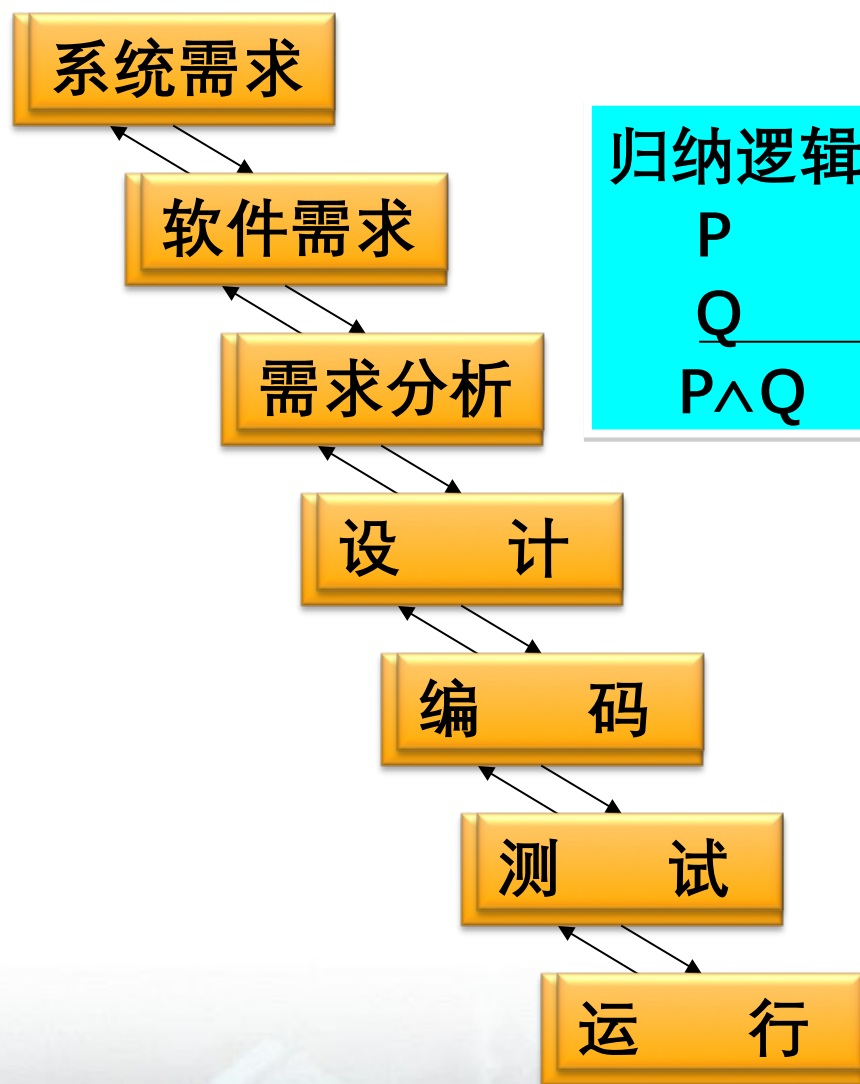
# ---- 软件生存周期模型

## 2 ) 瀑布模型

1970 年, W.Royce

(1) 瀑布模型将软件生存周期的各项活动规定为依固定顺序而连接的若干阶段工作;

(2) 瀑布模型规定了每一阶段的输入, 以及本阶段的工作成果, 作为输出传入下一阶段。



归纳逻辑:

P

Q

$P \wedge Q$



北京大学

## ---- 软件生存周期模型

- (1) 项目的开发依次经过：需求、设计、编码和单元测试、集成以及维护 –这一基本路径。
- (2) 通过每一阶段，提交以下产品：软件需求规约、设计文档、实际代码、测试用例、最终产品等。工作产品（又称可提交的产品， Deliverables ） 流经“正向”开发的基本步骤路径。
- (3) “反向”步骤流表示对前一个可提交产品的重复变更（又称为“返工” (Rework) ） 。
  - 由于所有开发活动的非确定性，因此是否需要重复变更，这仅在下一个阶段或更后的阶段才能认识到。
  - 返工不仅在以前阶段的某一地方需要，而且对当前正在进行的工作也是需要的。



### 关于瀑布模型的几点说明

#### (1) 瀑布模型的优点

虽然瀑布模型是一个比较“老”的、甚至过时的开发模型，但其优点为：

- ❶ 在决定系统怎样做之前，存在一个需求阶段，鼓励对系统“做什么”进行规约（即设计之前的规约）。
- ❷ 在建造构件之前，存在一个设计阶段，鼓励规划系统结构（即编码之前的设计）。
- ❸ 在每一阶段结束时进行复审，允许获取方和用户的参与。
- ❹ 前一步工作产品可作为下一步被认可的、文档化的基线。允许基线和配置早期接受控制。



[illegible]

A cartoon illustration of a man with brown hair, wearing a white short-sleeved shirt and a red tie. He is sitting at a light brown desk, leaning forward with his head resting on his right hand, looking directly at the viewer with a weary expression. His left hand is resting on the desk near some papers. A black wristwatch is visible on his left wrist.

**2**

[illegible]

1 □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ , □ □ □ □ □ □ □ □ □ □



北京大学



### (3) 瀑布模型适用的情况

- ① 需求已被很好地定义和理解；
- ② 过程设计人员也很清楚：开发组织非常熟悉为实现这一模型所需要的过程（或经过培训后，熟悉什么时候来支持这一项目，以实现这一模型所需要的过程）。

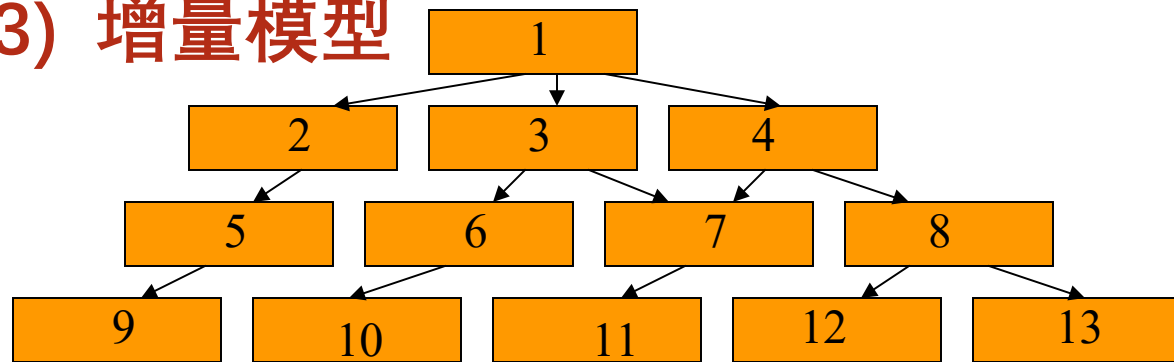
因此为了避免产生过多的反复迭代工作，增加开发成本，一般在准备采用瀑布模型（也包括其他模型）时，需要考虑以下 2 个问题：

- (1) 过程设计人员必须对初始产品（通常是软件需求规约，SRS）的不确定性进行评估。
- (2) 组织是否具有熟练实施每个活动和任务的历史经验。



# ---- 软件生存周期模型

## 3) 增量模型



增量 1

{ 1,2,5,9 }

增量 2

{ 3,6,7,4,10,11 }

增量 3

{ 8,12,13 }

增量  
1

增量  
2

增量  
3

管理

增量规约

增量设计

纠错性分析

增量实现

该模型有一个假设，即需求可以分段，成为一系列增量产品，每一增量可以分别地开发。



北京大学





## ---- 软件生存周期模型



关于增量模型的几点说明：

### (1) 增量模型的优点

作为瀑布模型的第一个变体，具有瀑布模型的所有优点。此外，它还有以下优点：

- ① 第一个可交付版本所需要的成本和时间是很少的；
- ② 开发由增量表示的小系统所承担的风险是不大的；
- ③ 由于很快发布了第一个版本，因此可以减少用户需求的变更；
- ④ 允许增量投资，即在项目开始时，可以仅对一个或两个增量投资。

注：如果采用增量投资方式，那么客户就可以对一些增量进行招标。

然后，开发人员按提出的截止期限进行增量开发，这样客户就可以用多个契约来管理组织的资源和成本。



北京大学

( 2 ) 缺点：



如果增量模型不适于某些项目，或使用有误，则有以下缺点：

- ① 如果没有对用户的变更要求进行规划，那么产生的初始增量可能会造成后来增量的不稳定；
- ② 如果需求不像早期思考的那样稳定和完整，那么一些增量就可能需要重新开发，重新发布；
- ③ 管理发生的成本、进度和配置的复杂性，可能会超出一些组织的能力。



### ( 3 ) 该模型的适用情况

① 在开始开发时，需求很明确，且产品还可被适当地分解为一

些独立的、可交付的软件（构造增量：Build increments . 若一个增量不需要交付给客户的话，那么这样的增量通常称为一个“构造” (Build) 。若增量需要被交付的话，那么它们就被认为是发布版本 (Released version) 。 ) ；

② 在开发中，期望尽快提交其中的一些增量产品。

例如：

一个数据库系统，它必须通过不同的用户界面，为不同类型的用户提供不同的功能。在这一情况下，首先实现完整的数据库设计，并把一组具有高优先级的用户功能和界面作为一个增量；以后，陆续构造其它类型用户所需求的增量。

## ---- 软件生存周期模型



**Microsoft®**

附：微软“同步 - 稳定的产品开发模型”

- 将项目分为若干个里程碑阶段
  - 定义稳定、灵活的体系结构，并为构件和子系统的开发提供统一的接口
  - 开发构件，维持一个可发布的系统版本
- 可以准确把握项目进展情况
- 增强开发人员的信心和成就感
- 可以随时根据市场情况及时作出调整



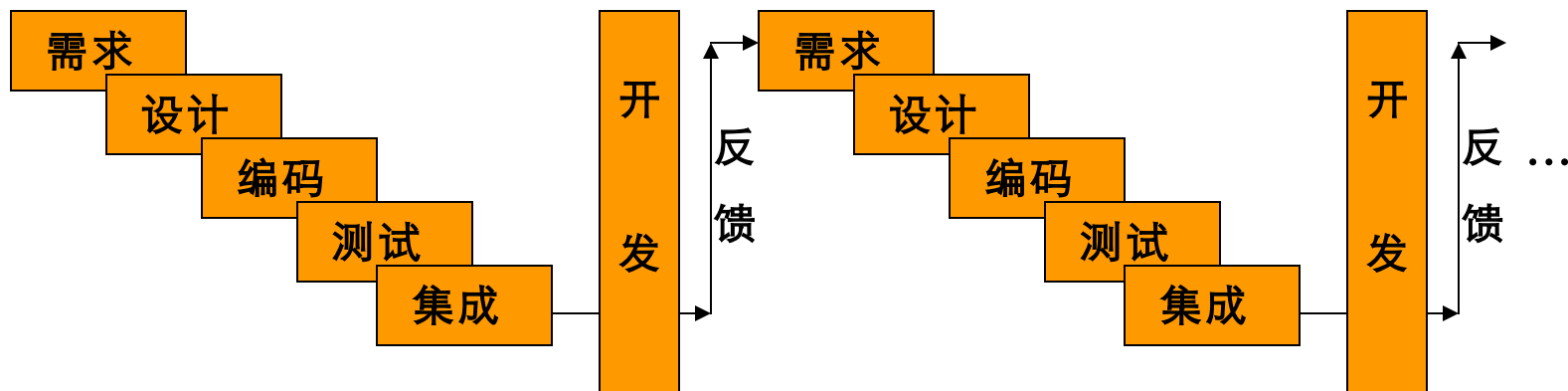
北京大学



□ ) □ □ □ □ □ **Evolutionary model** □

--	--	--	--	--

- □□□□□□□□□□□□□□□□
- □□□□□□□□□ , □□□□□□
- □□□□□□□□ , □□□□□□□□



## 核心系统开发

## 第二次迭代



北京大学



## ---- 软件生存周期模型

### 关于演化模型的几点说明

#### (1) 主要特征

该模型显式地把增量模型扩展到需求阶段。由图可以看出，为了第二个构造增量，使用了第一个构造增量来精化需求。这一精化可以有多个来源和路径：

首先，如果一个早期的增量已向用户发布，那么用户会以变更要求的方式提出反馈，以支持以后增量的需求开发。

第二，通过实实在在地开发一个构造增量，为以前还没有认识到的问题提供了可见性，以便实际地开始这一增量的工作。





### (2) 与瀑布模型的关系

在演化模型中，仍然可以使用瀑布模型来管理每一个演化的增量。一旦理解了需求，就可以像实现瀑布模型那样开始设计阶段和编码阶段。

### (3) 使用演化模型应注意的问题

不能弱化需求分析阶段的工作。其原因是：在项目开始时，需要考虑所有可能需求源的重要性的风险，并对这些需求源的可用性进行评估，只有这样才能识别和界定不确定的需求，并识别第一个增量中所包含的需求。



### (4) 演化模型的长处和不足

其长处与增量模型是类似的，但还具有以下优点：

- ① 在需求不能予以规约时，可以使用这一演化模型。
- ② 用户可以通过运行系统的实践，对需求进行改进。
- ③ 与瀑布模型相比，需要更多用户 / 获取方的参与。

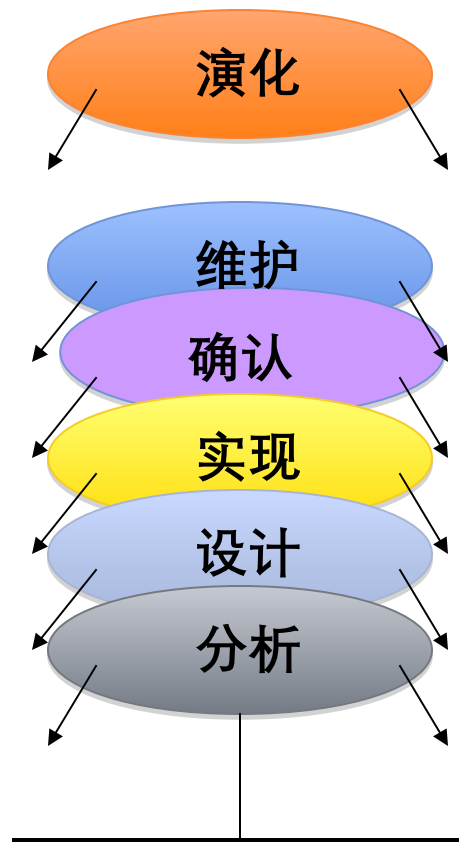
不足有：

- ① 演化模型的使用需要有力的管理。
- ② 演化模型的使用很容易成为不编写需求或设计文档的借口，即使很好地理解了需求或设计。
- ③ 用户 / 获取方不易理解演化模型的自然属性，因此当结果不够理想时，可能产生抱怨。



## 5 ) 喷泉模型

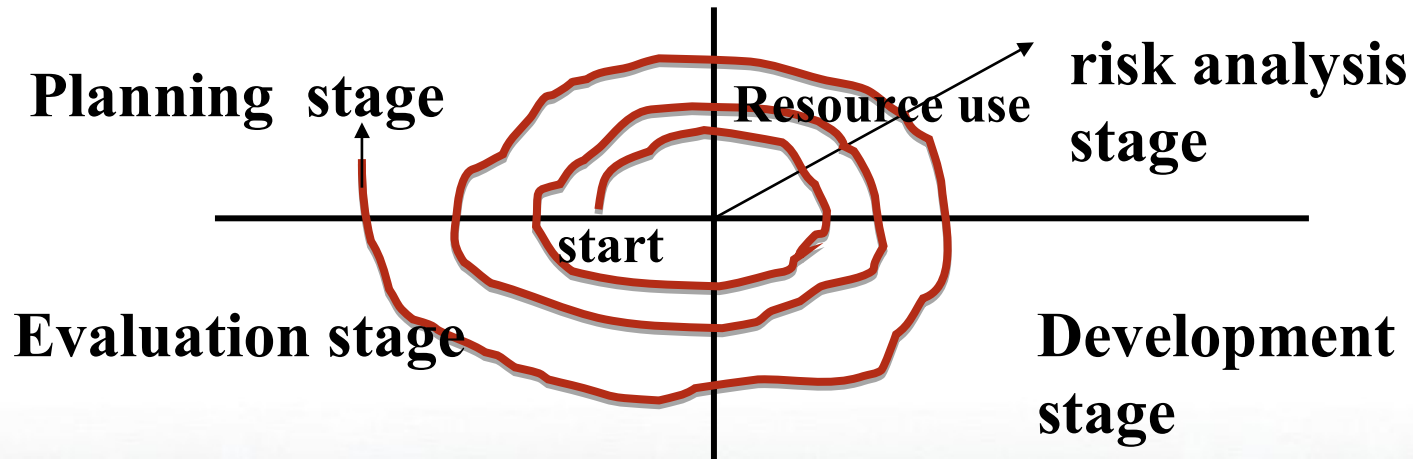
- 特征：迭代  
无缝
- 与面向对象技术的  
关系





□ □ □ □ **Dr. Barry Boehm**[Boehm 1988] □ □ □ □ □ □ □ □ □ □

□ □



□ □ □ □ □ “ □ □ ” □ □ □



北京大学



## ---- 软件生存周期模型

其中：

- ① 评估和风险分析阶段都可作出一个决策：项目是否继续。
- ② 螺旋循环的次数指示了已消耗的资源；
- ③ 在规划阶段、风险分析阶段和开发阶段均进行需求规约活动；
- ④ 在早期螺旋循环中，为了给最终的实现给出一些指导性决策，经常使用原型构造；
- ⑤ 设计和实现活动一般是在开发阶段进行；
- ⑥ V&V 活动在开发阶段和评估阶段进行；





## ---- 软件生存周期模型

关于螺旋模型的几点说明：

(1) 该模型关注解决问题的基本步骤：

- ◆ 标识问题；
- ◆ 标识一些可选方案，选择一个最佳方案；
- ◆ 遵循动作步骤，并实施后续工作。

其中只要完成了开发的一个迭代，开发的另一个迭代就开始。

(2) 螺旋模型的一个特征是，实际上只有一个迭代过程真正开发可交付的软件。因此，如果

- ◆ 项目的开发风险很大，或
- ◆ 客户不能确定系统需求，在更广泛的意义上来讲，还包括系统或系统类型的要求，这时螺旋模型就是一个好的生存周期模型。



北京大学



### (3) 与其它模型的关系

① 与演化模型一样，螺旋模型也使用瀑布模型作为一个嵌入的过程 - 即分析、设计、编码、实现和维护的瀑布过程，是螺旋一周的组成部分。



② 尽管螺旋模型和一些迭代模型在框架和全局体系结构方面是等同的，但所关注的**阶段**以及它们的**活动**是不同的。

具体地说：① 标识客户想要的是一个什么样的系统；② 确定风险和效益的可选路线；③ 选择最优方案；④ 开发系统；⑤ 评估完成情况等；⑥ 重新开始。

即 螺旋模型扩展了增量模型的管理任务范围。而增量模型是基于以下假定：需求是最基本的、并且是唯一的风险源。而在螺旋模型中，决策和降低风险的空间是相当广泛的。



## 7) 快速原型模型

快速原型是快速建立起来的可以在计算机上运行的程序，它所能完成的功能往往是最终产品能完成的功能的一个子集。原型在开发中的作用：

(1) 获得用户的真正需求

(2) 可用于为一个项目或项目中某些部分，确定技术、成本和进度的可能性

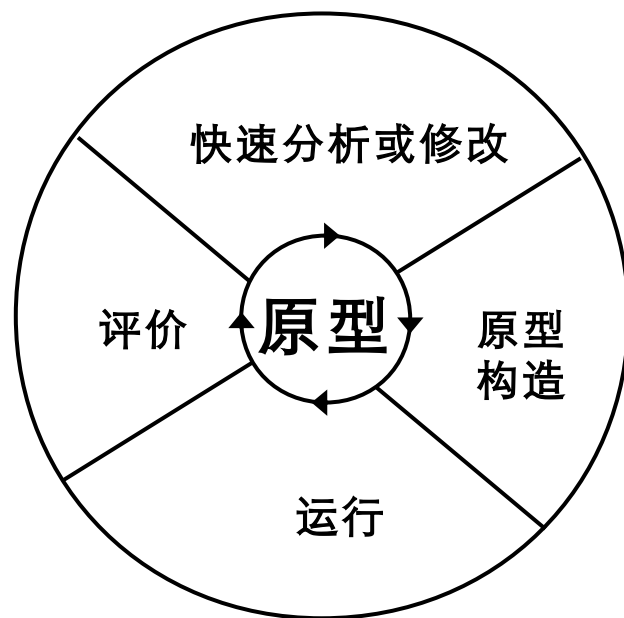
快速原型模型中各主要部分：

### 1. 原型快速分析

指在分析者和用户的紧密配合下，快速确定软件系统的基本要求。

### 2. 原型构造

在原型分析的基础上，根据基本需求规格说明，忽略细节，只考虑主要特性，快速构造一个可运行的系统。



快速原型模型



### 3. 原型运行与评价

软件开发人员与用户频繁通信、发现问题、消除误解的重要阶段，目的是发现新需求并修改原有需求。

### 4 . 原型修正

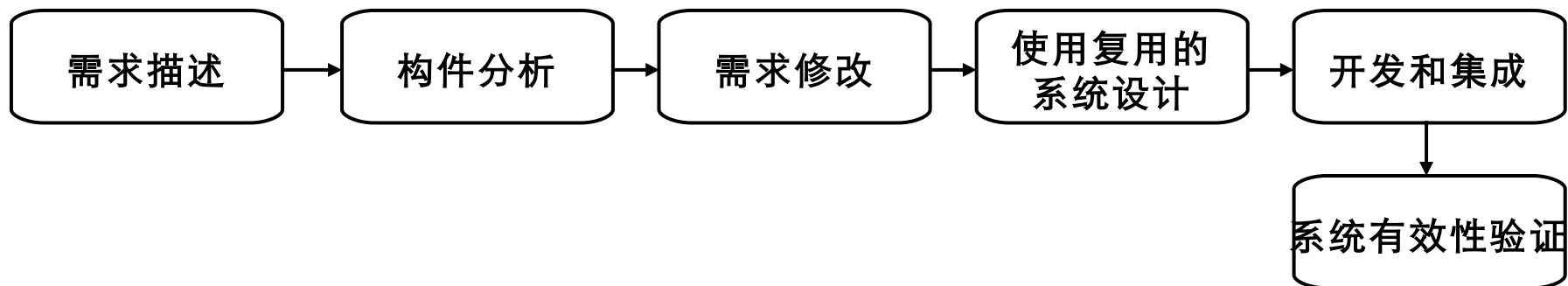
对原型系统，要根据修改意见进行修正。

### 5 . 判定原型完成

如果原型经过修正或改进，获得了参与者的一致认可，那么原型开发的迭代过程可以结束



## 8) 面向复用的软件开发模型



面向复用的软件开发模型依赖于存在大量可复用的软件构件以及能组合这些构件的集成框架。

有三种类型的软件构件可用于面向复用的软件开发过程：

1、通过标准服务开发的 Web 服务，可用于远程调用。

2、对象的集合，作为一个包和构件框架，如 .NET 或者 J2EE 等集成在一起。

3、独立的软件系统，通过配置在特定的环境下使用。





## (1) 面向复用的软件开发模型的优点

- ① 减少了需要开发的软件数量，从而降低了软件开发成本，以及降低了开发中的风险。
- ② 可使软件快速地交付。

## (2) 面向复用的软件开发模型的缺点

- ① 需求妥协是不可避免的，这可能导致一个不符合用户真正需要的系统。
- ② 对系统进化的控制将失效，因为可复用的构件新版本可能是不受机构控制的。

