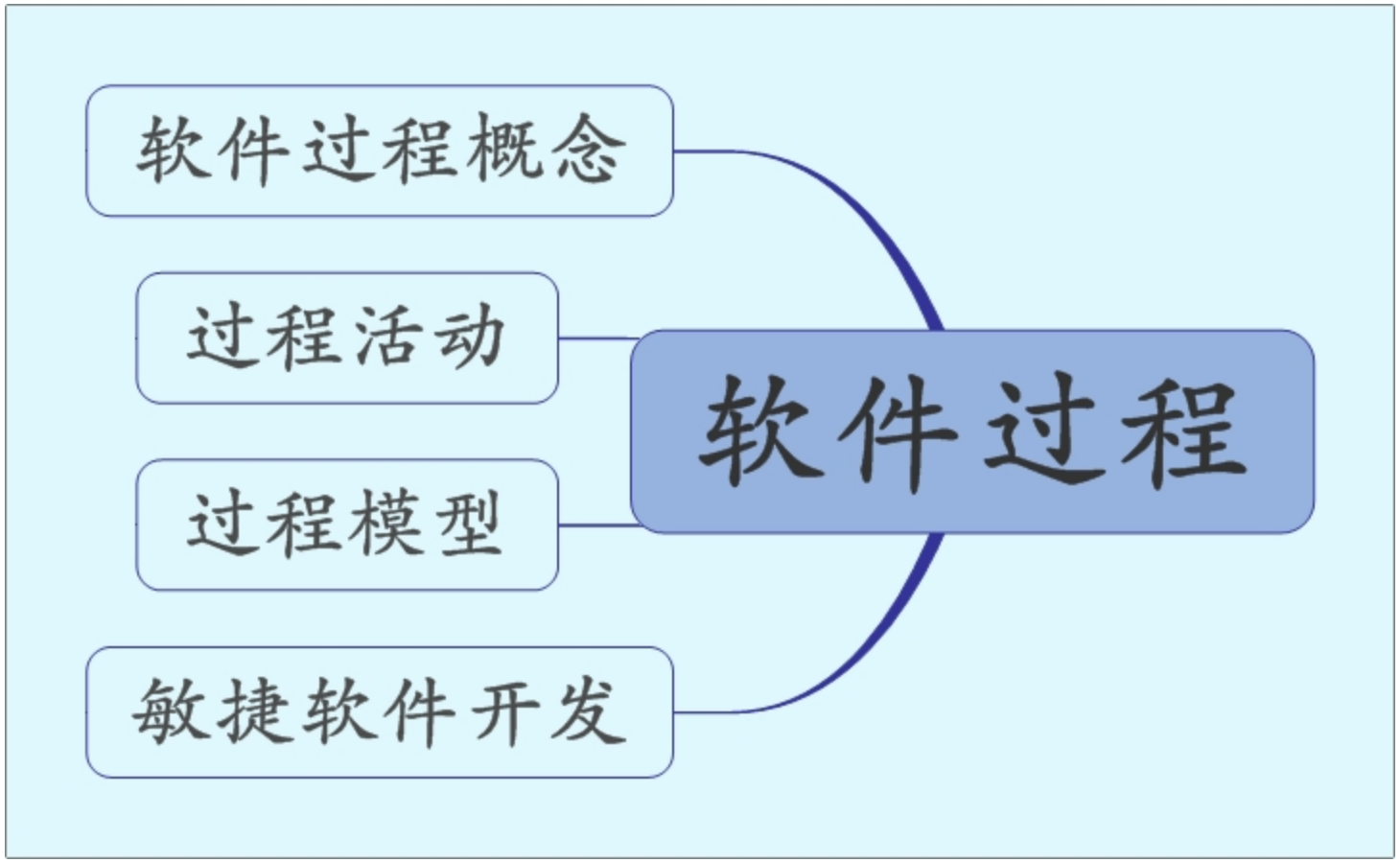
outline



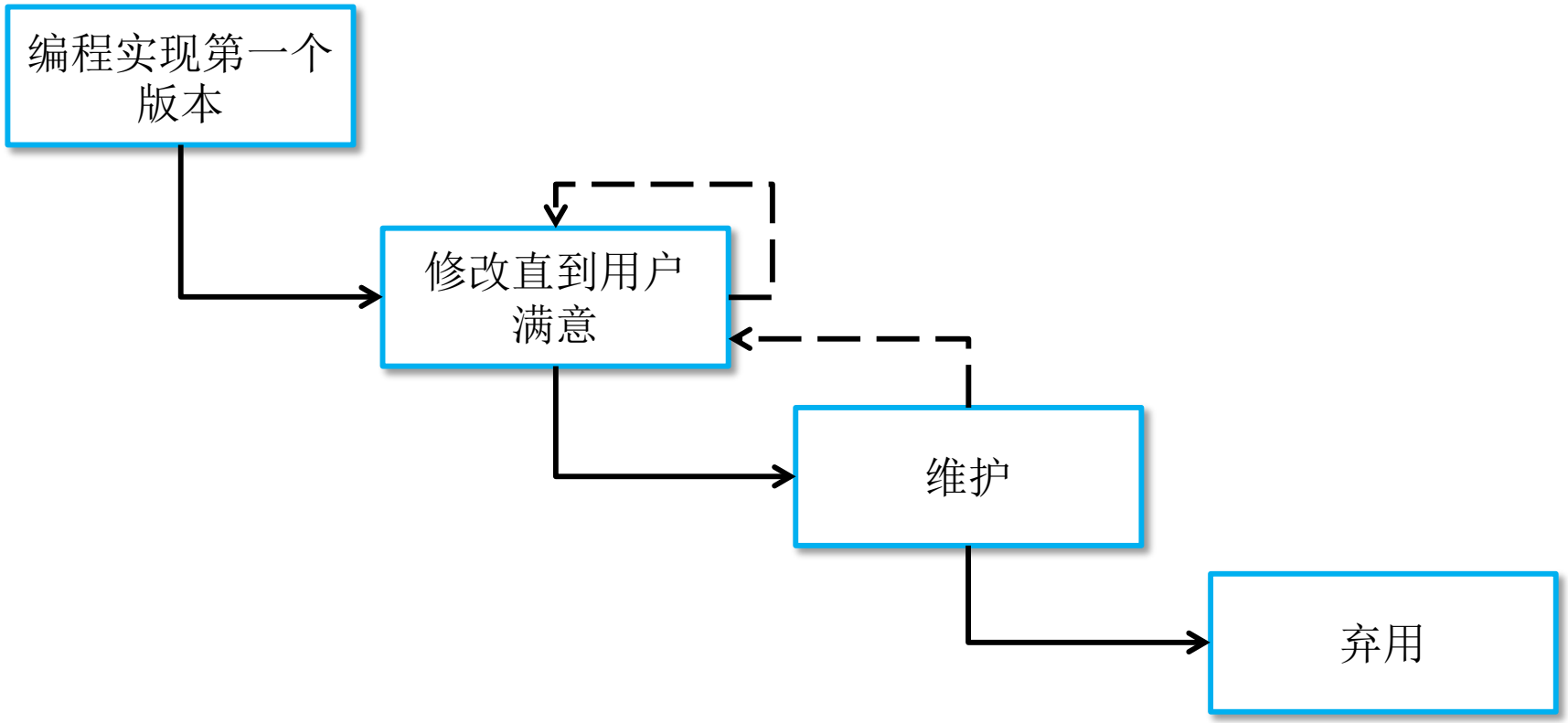
软件过程模型

- 软件过程模型是一个软件过程的抽象表示（路线图）。
- 为了改变软件开发的混乱状况，使软件开发更加有序。

- **建造-修正模型**
- **瀑布模型**
- **快速原型模型**
- **增量模型**
- 螺旋模型
- 喷泉模型
- 形式化方法模型
- 基于构件的开发模型
- **RUP**
- **敏捷过程与极限编程**
- 微软过程模型
- ……

建造－修正模型(Code-and-Fix)

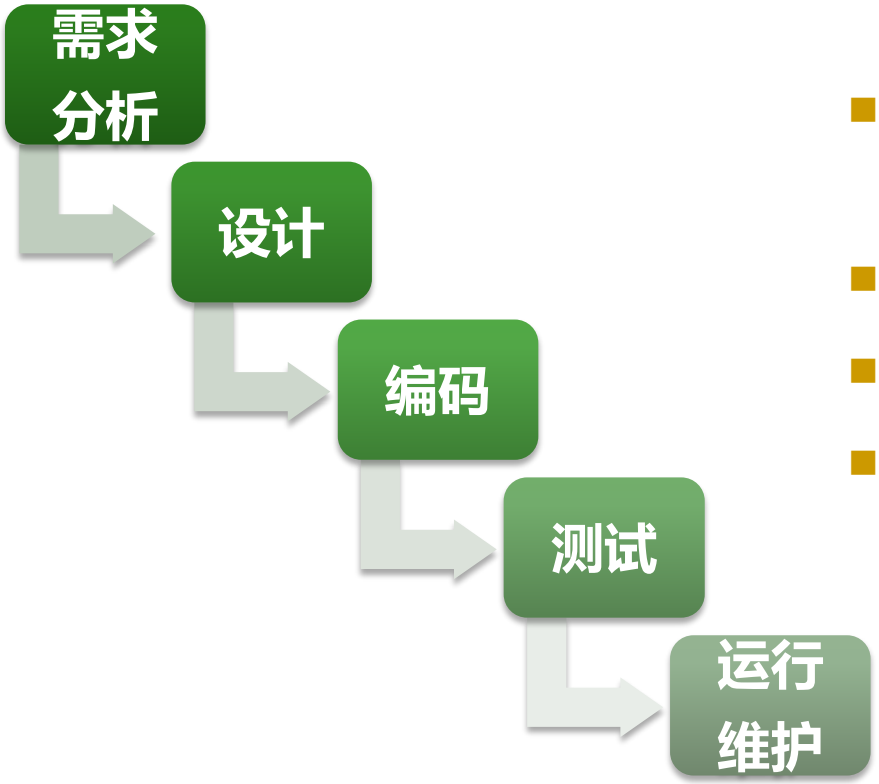
- 没有需求分析、设计
- 直接编码



建造—修正模型

- **优点：**
  - 简单
- **缺点：**
  - 对于规模稍大的项目，采用这种模型很危险。
  - 难以维护
- 适用于小规模（200行左右）且无需维护的程序，不适用于稍大规模的系统。

瀑布模型（Waterfall Model）



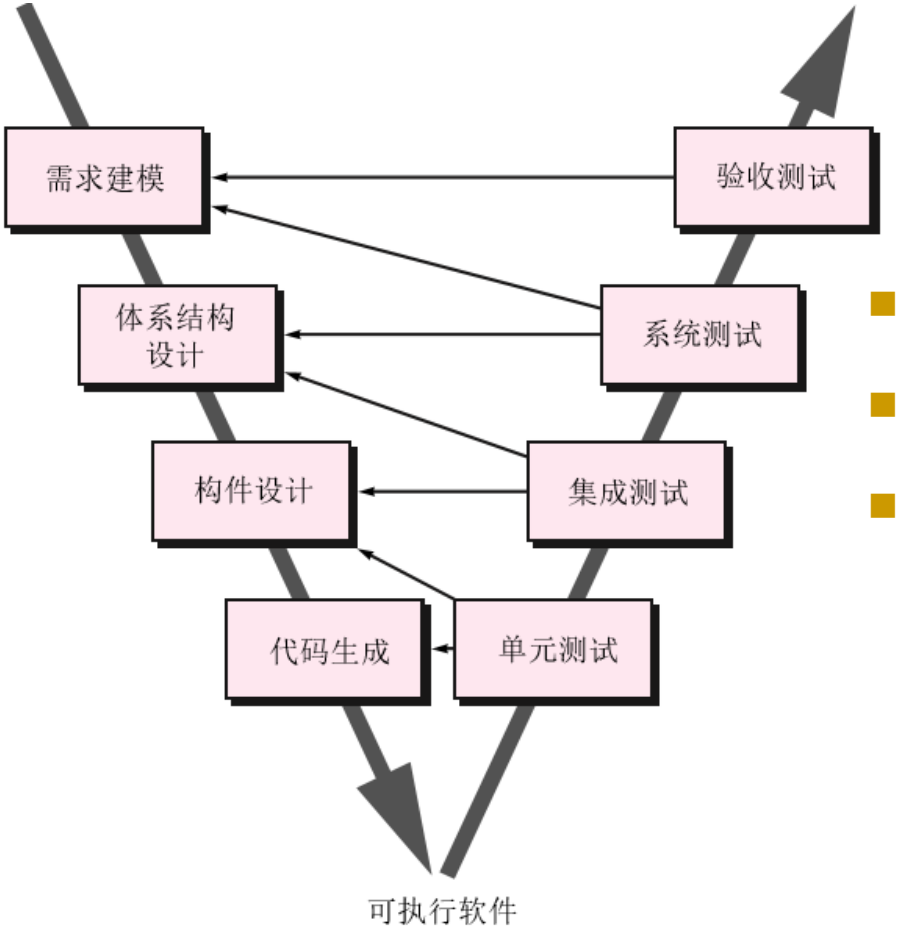
- 系统的、规范的、顺序的软件开发方法
- 每个阶段都有里程碑和可交付产品
- 推迟实现

我们可以看得到瀑布模型在编码前进行了大量的准备工作。即推迟物理实现，这样做可以使我们对项目有深入的思考，提高了我们写代码的效率以及代码质量

- 适用于需求已准确定义和相对稳定的项目
  - 对一个已经存在的系统进行明确定义的适应性调整或增强
  - 新开发的系统

缺点：
1）开发过程一般不能逆转，否则代价太大；
2）实际的项目开发很难严格按该模型进行；
3）客户往往很难清楚地给出所有的需求，而该模型却要求如此。
4）软件的实际情况必须到项目开发的后期客户才能看到，这要求客户有足够的耐心。

改进的瀑布模型



V模型

- 瀑布模型的变体
- 测试活动与分析和设计相联系
- V模型使得隐藏在瀑布模型中的重做和迭代活动更加明确
  - 如果在单元测试期间发现了问题，则需要重新执行左边的构件设计和代码生成

问题

- 实际的软件项目很少遵守瀑布模型提出的顺序
- 客户通常难以清楚地描述所有的需求
- 客户要等待很长时间才能够得到可运行的系统

- 解决办法
  - 开始就给用户展示一个目标系统的的雏形，让用户使用，然后逐步进行修改，直至成功。
  - **原型开发**