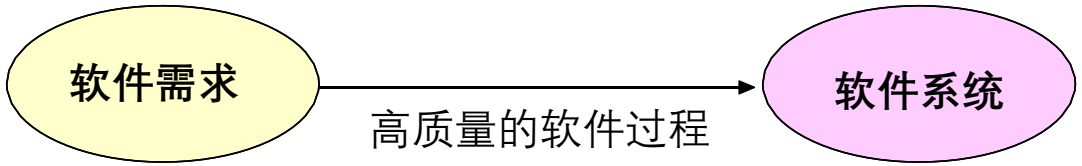
前言：软件工程概论

- 软件工程是一门研究用工程化方法构建和维护高质量软件的学科。
- 软件产品：文档、数据、程序的集合
- 软件开发过程：软件产品的制造过程
- 必须把二者结合起来去考虑，而不能忽略其中任何一方。



软件过程重要性

- 软件就像精密的机器，需要各部件完美的配合才能使用。随着软件越来越复杂，人们越来越重视软件过程的重要性。

第二部分 软件项目开发过程与管理

- 2.1 软件项目开发过程
- 2.2 软件项目开发管理

2.1 软件项目开发过程

- 软件生命周期、软件过程模型(软件生命周期模型)的概念及其关系
- 明白软件开发过程的典型阶段
- 典型软件过程模型

  - 瀑布模型、
  - 增量过程模型
    - 增量模型
    - 快速应用程序开发（RAD）
  - 演化过程模型
    - 快速原型开发模型
    - 螺旋模型

- 对比分析几种典型的软件过程模型的异同

软件生命周期

- 同任何事物一样，一个软件产品或软件系统也会经历孕育、诞生、成长、成熟、衰亡这一过程。
- 软件生命周期把上述过程划分为若干阶段，使得每个阶段有明确的任务。
- 软件从提出开发到最终灭亡所经历的时期，不同阶段的特征、任务、产品、所需技术不一样。

软件过程模型（软件生命周期模型）

软件过程模型：

最早提出过程模型是为了改变软件开发的混乱状况，使软件开发更加有序。

为软件工程工作提供了特定的路线图，该路线图规定了所有活动的流程、动作、任务、迭代的程度、工作产品及要完成的工作应如何组织。

2.1 软件项目开发过程

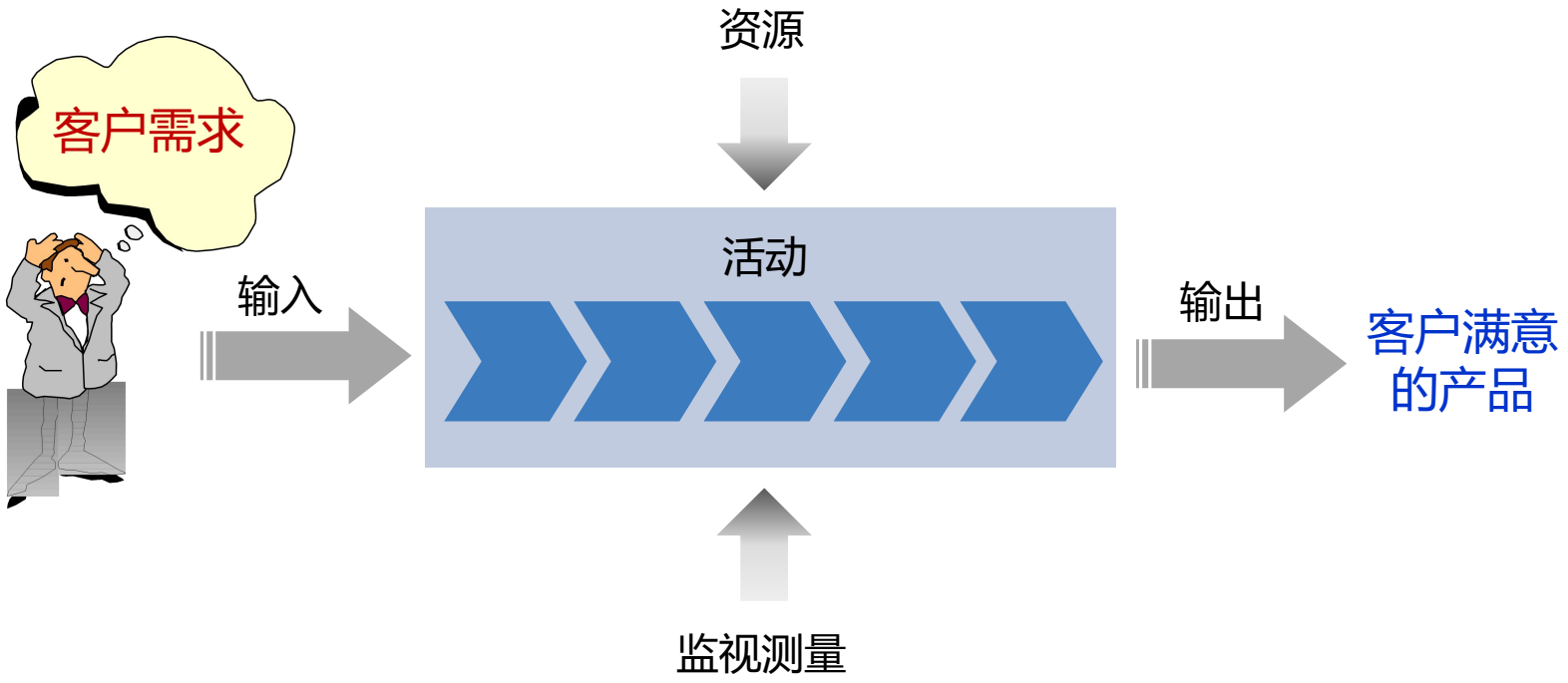
- 软件生命周期、软件过程模型(软件生命周期模型)的概念及其关系
- 明白软件开发过程的典型阶段
- 典型软件过程模型

  - 瀑布模型、
  - 增量过程模型
    - 增量模型
    - 快速应用程序开发（RAD）
  - 演化过程模型
    - 快速原型开发模型
    - 螺旋模型

- 对比分析几种典型的软件过程模型的异同

软件开发过程方法

过程是一组将输入转化为输出的相互关联或相互作用的活动；过程方法是系统地识别和管理组织内所使用的过程，保证更有效地获得期望的结果。

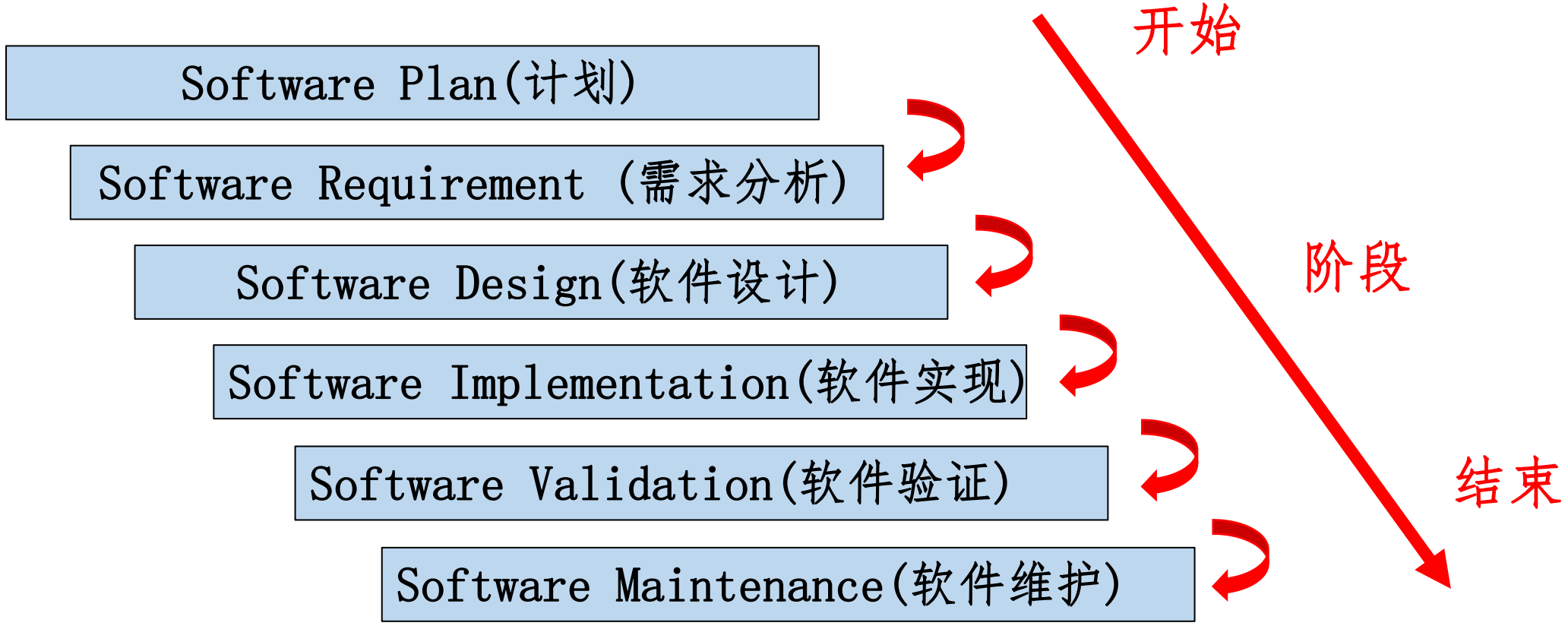


为什么要将软件开发过程划分为多个阶段？

- 六七十年代的软件开发过程：程序员个人设计、个人操作、小作坊式的开发过程，开发的软件可靠差，难以维护，无法满足快速发展的软件需求，因而产生软件危机。

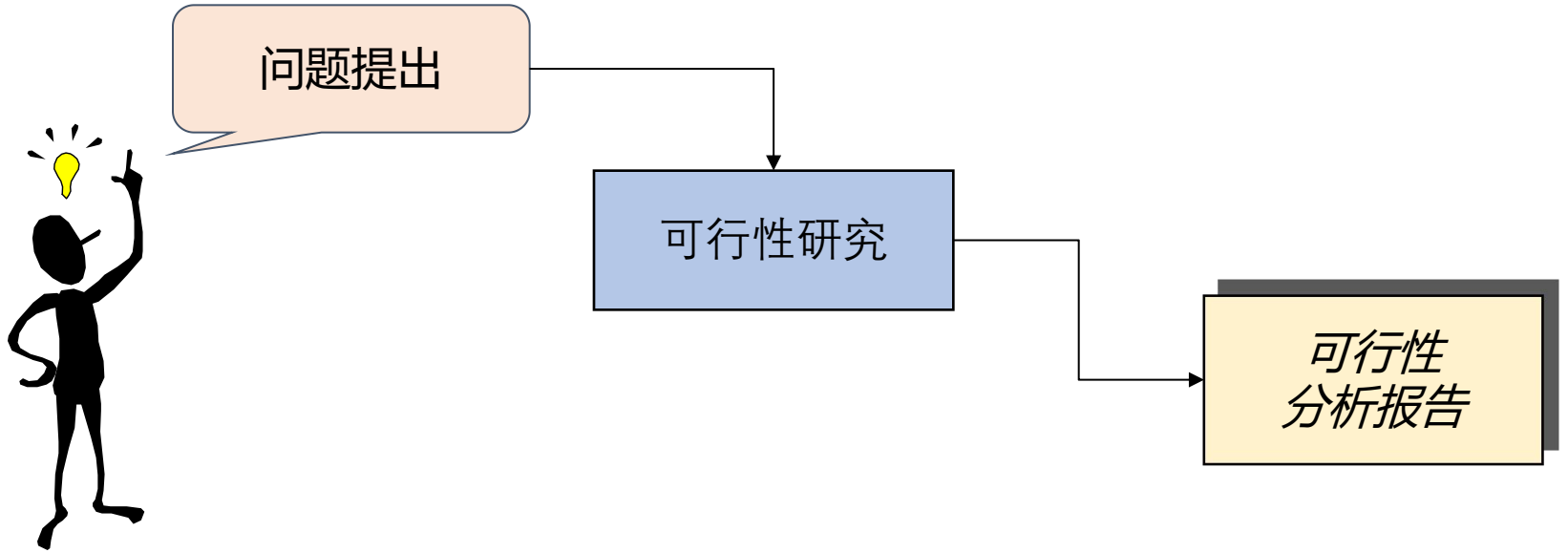
- 如何解决软件危机？
  - 把软件开发过程划分为多个阶段使得每个阶段有明确的任务，通过问题的分解，将大规模、结构复杂的软件开发变的容易控制和管理。

软件开发过程的典型阶段



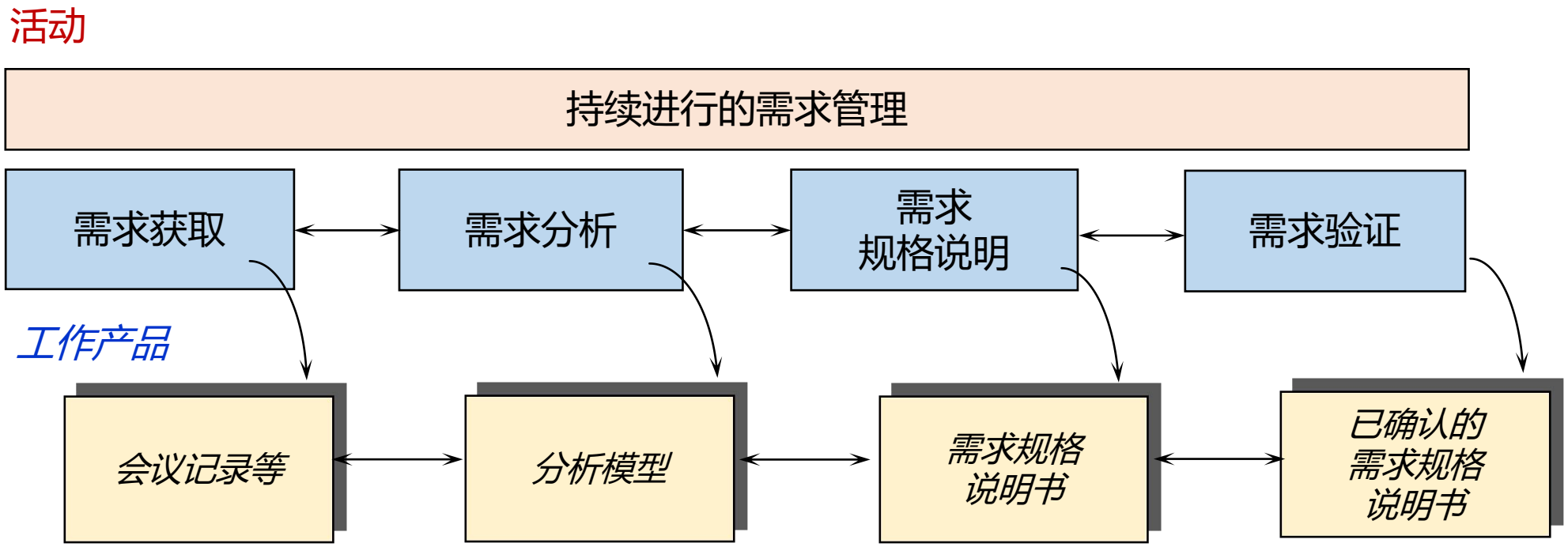
软件开发过程-计划

问题定义：人们通过开展技术探索和市场调查等活动，研究系统的可行性和可能的解决方案，确定待开发系统的总体目标和范围。



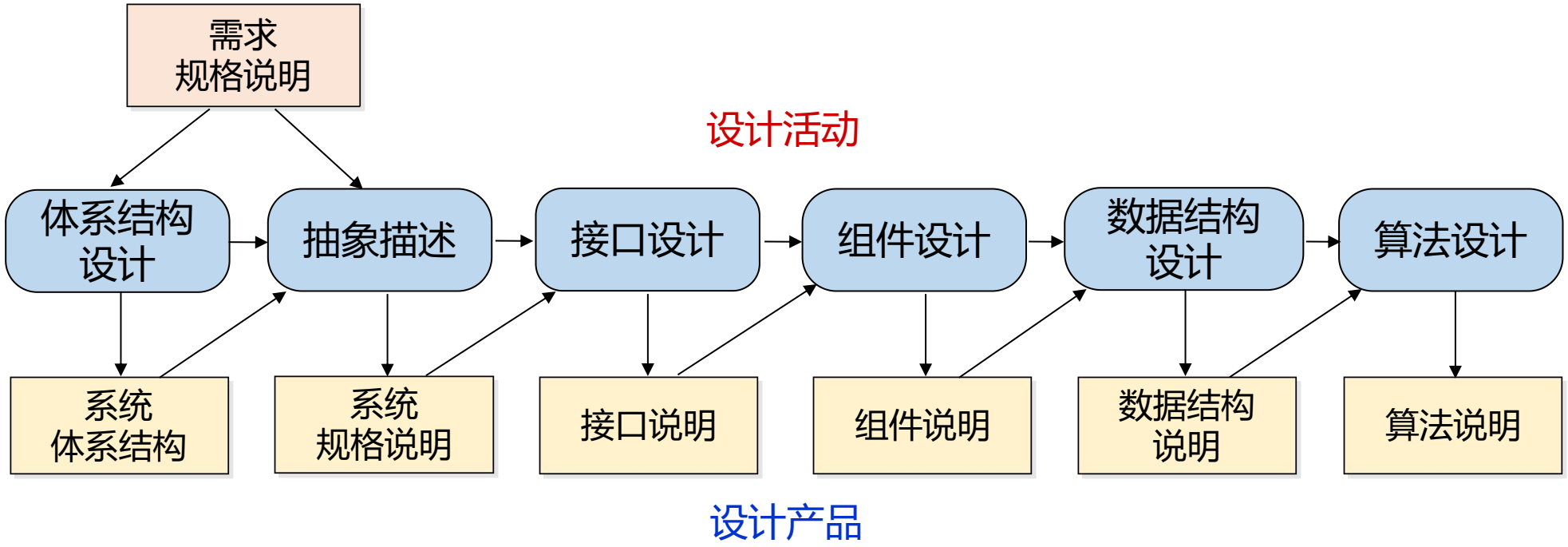
软件开发过程-需求分析

需求分析：在可行性研究之后，分析、整理和提炼所收集到的客户需求，建立完整的需求分析模型，编写软件需求规格说明。



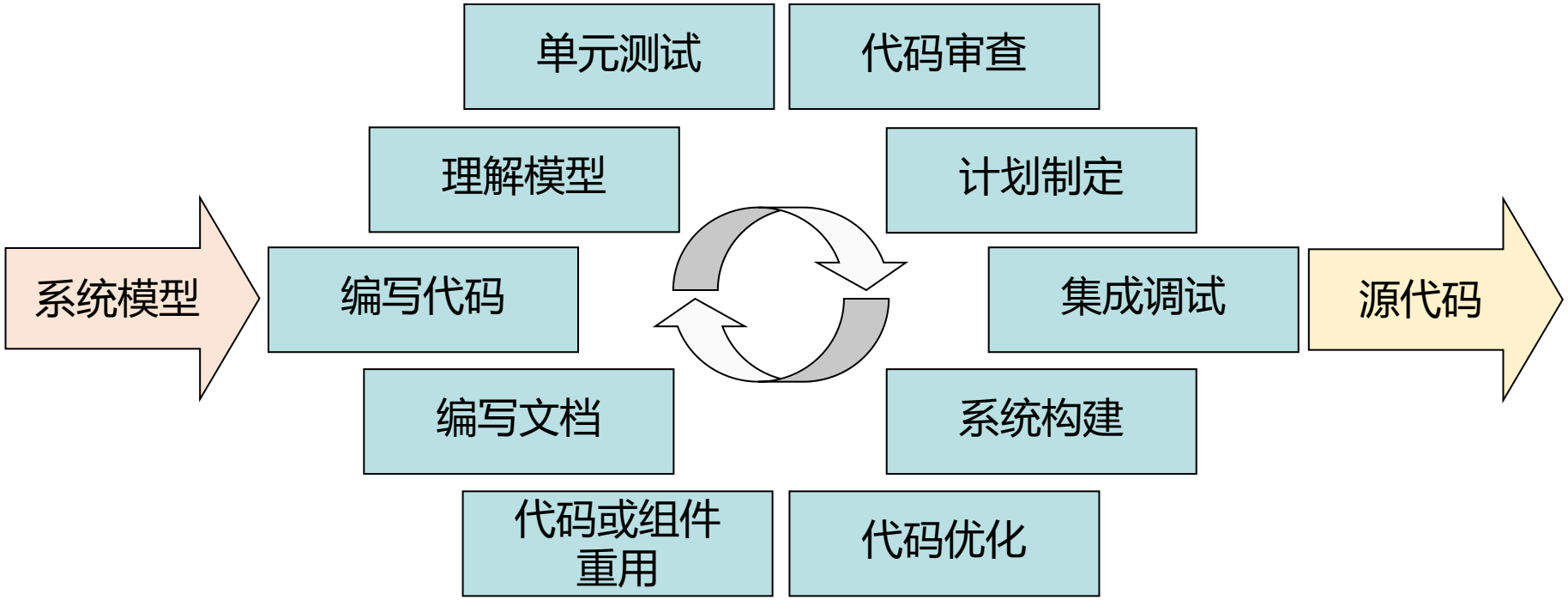
软件开发过程-软件设计

软件设计：根据需求规格说明，确定软件体系结构，进一步设计每个系统部件的实现算法、数据结构及其接口等。



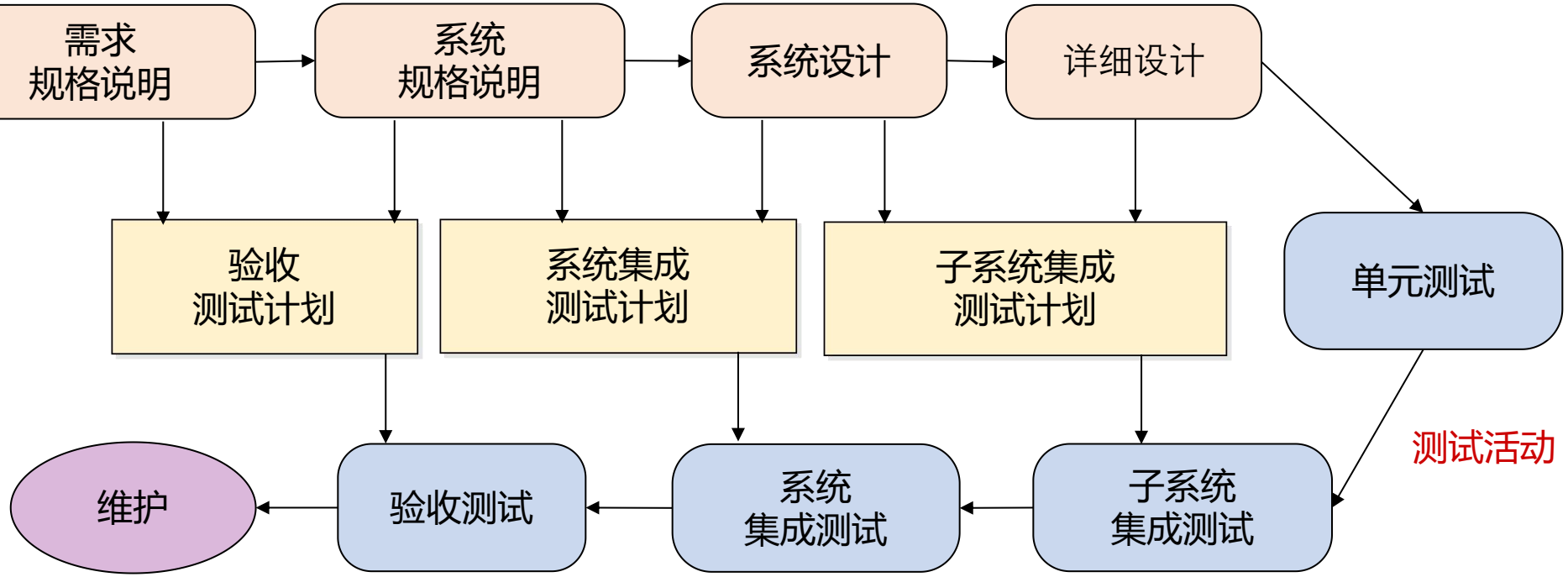
软件开发过程-软件实现

软件实现：概括地说是将软件设计转换成程序代码，这是一个复杂而迭代的过程，要求根据设计模型进行程序设计以及正确而高效地编写和测试代码。



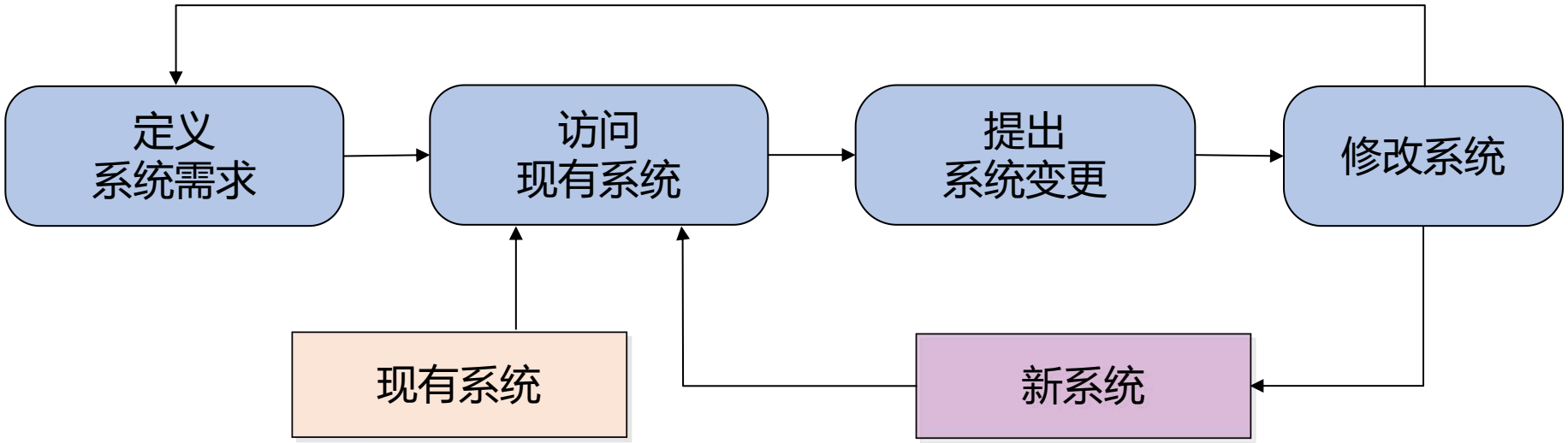
软件开发过程-软件验证

软件验证：检查和验证所开发的系统是否符合客户期望，包括单元测试、子系统测试、集成测试和验收测试等。



软件开发过程-软件维护

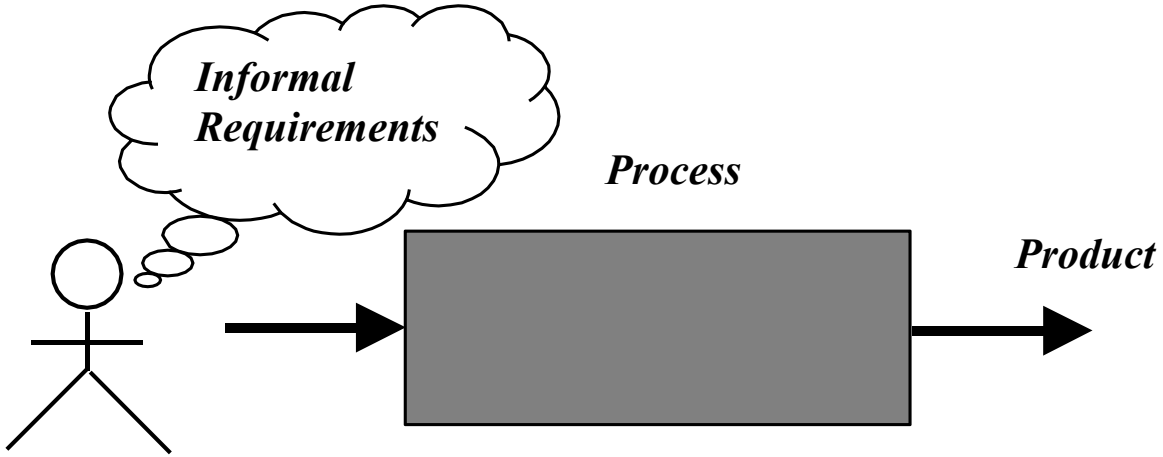
软件维护：系统投入使用后对其进行改进，以适应不断变化的需求。完全从头开发的系统很少，将软件系统的开发和维护看成是一个连续过程更有意义。



注意

- 各阶段或轻或重，视具体情况而定。
- 每个阶段的产物需要验证和确认。

为什么要验证和确认？



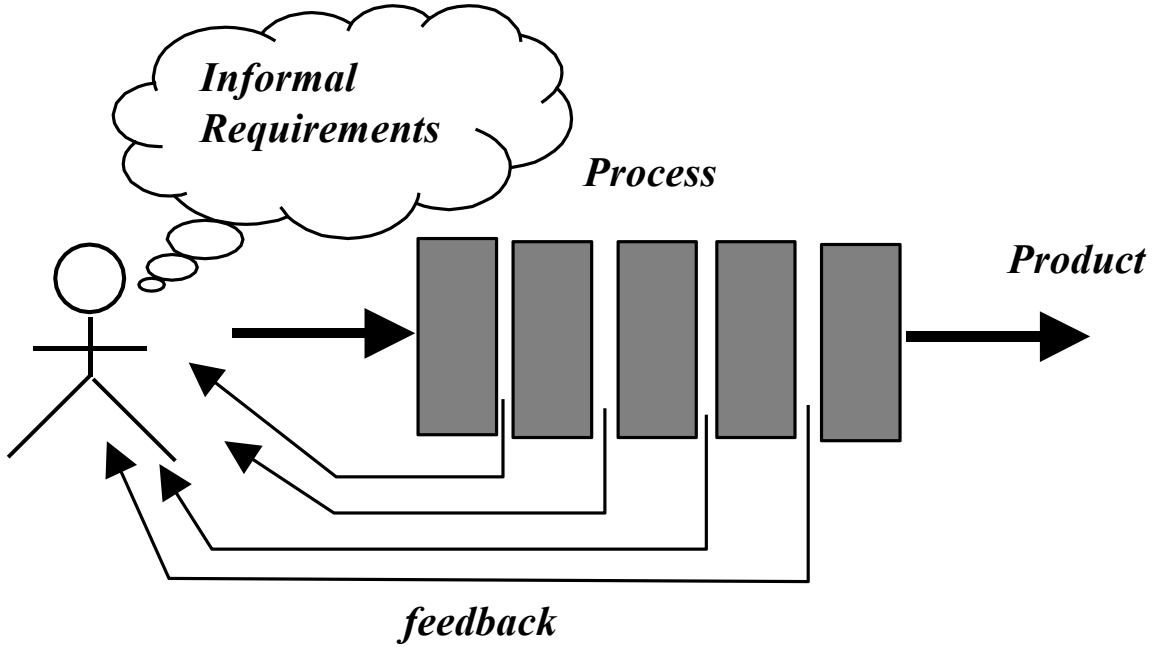
- **存在的问题：**
  - 要求开发之前需求被充分理解
  - 与客户的交互只在开始(需求)和最后(发布)
  - 实际情况？用户的需求是模糊的

…构造一所房子…

相同的目标

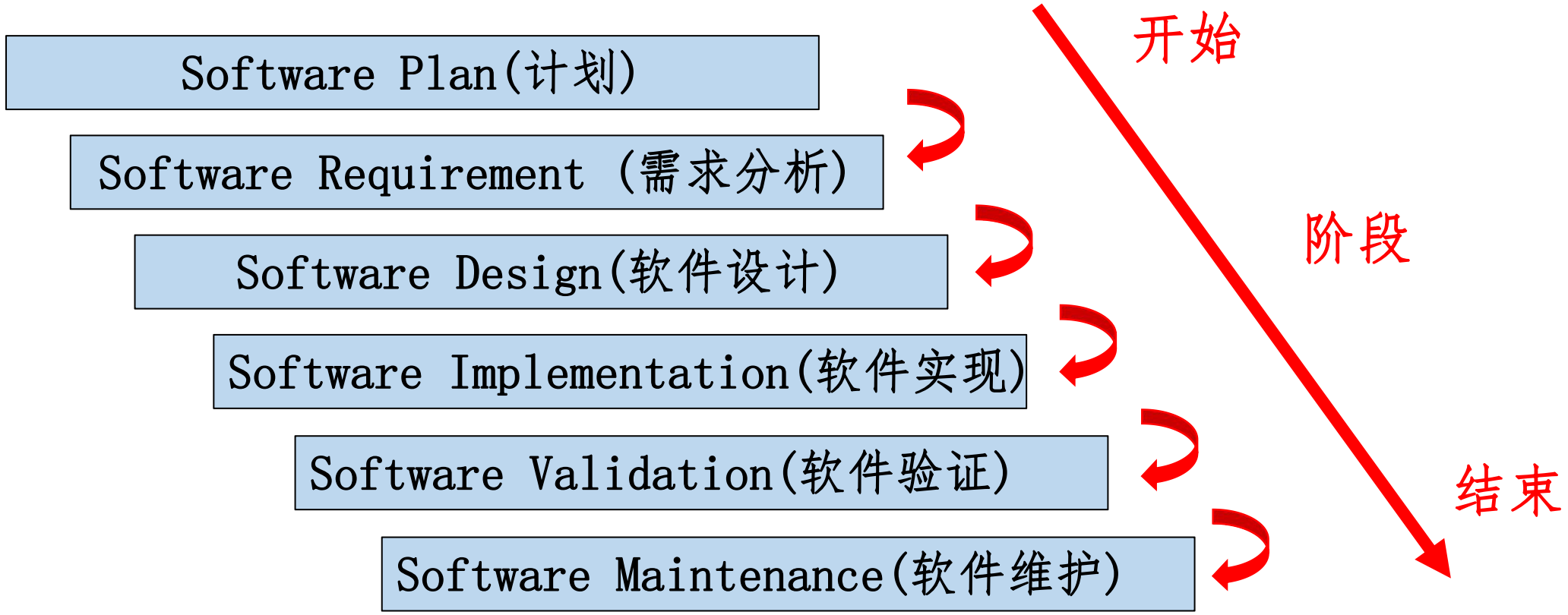
活动

为什么要验证和确认？



- 优点：
  - 通过用户的反馈进行验证和确认
  - 开发的产品与客户的需求接近

软件开发过程的典型阶段



- 可以通过调整软件开发不同阶段的顺序使之适应不同的情况。

2.1 软件项目开发过程

- 软件生命周期、软件过程模型(软件生命周期模型)的概念及其关系
- 明白软件开发过程的典型阶段
- 典型软件过程模型

  - 瀑布模型
  - 增量过程模型
    - 增量模型
    - 快速应用程序开发（RAD）
  - 演化过程模型
    - 原型开发模型
    - 螺旋模型

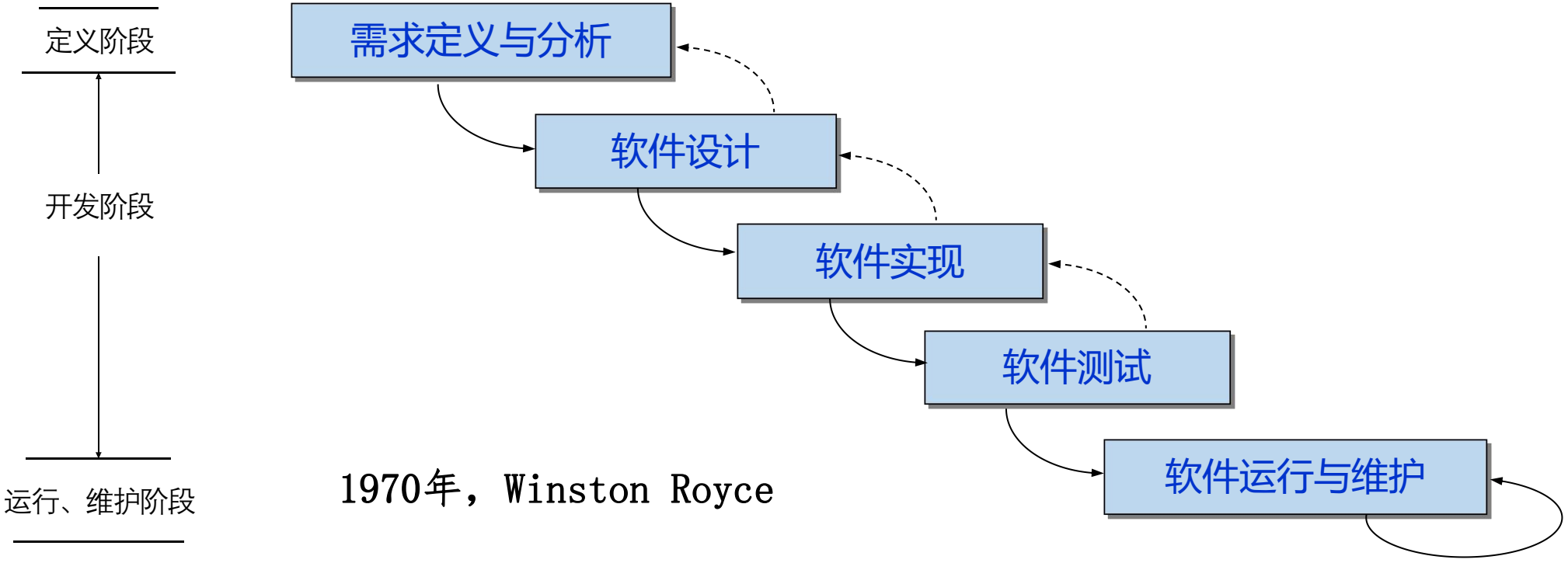
- 对比分析几种典型的软件过程模型的异同

典型软件过程模型

- 瀑布模型
- 增量过程模型
  - 增量模型
  - 快速应用程序开发（RAD）
- 演化过程模型
  - 快速原型开发模型
  - 螺旋模型

瀑布模型(Waterfall Model)

瀑布模型的开发阶段严格按照线性方式进行，每一个阶段具有相关的里程碑和交付产品，且需要确认和验证。



1970年，Winston Royce

瀑布过程模型思想和特点

- **基本思想**
  - 将软件开发过程划分为分析、设计、编码、测试等阶段
  - 工作以线性方式进行，上一阶段的输出是下一阶段的输入
  - 每个阶段均有里程碑和提交物
- **特点**
  - 需求最为重要，假设需求是稳定的
  - 以文档为中心，文档是连接各阶段的关键

瀑布过程模型思想和特点

讨论：瀑布模型是否反映了实际的软件开发过程？软件开发作为一个问题求解过程，应当具备什么特点？

瀑布模型产生于硬件领域，它是从制造业的角度看待软件开发的。制造业是重复生产某一特定的产品，而软件开发并不是这样的，随着人们对问题的逐步理解以及对可选方案的评估，软件在不断演化。因此，软件开发是一个创造的过程，而不是一个制造的过程。

瀑布模型带来的问题

- 文档多且完善
  - 项目建议书
  - 可行性分析报告
  - 需求分析报告
  - 概要设计报告
  - 系统架构设计报告
  - 数据库设计报告
  - 详细设计报告
  - 界面设计报告
  - 集成测试报告
  - 系统安装配置说明
  - 用户使用报告

  …… ……

- 文档的问题
  - 对于文档的评估需要各领域的专家，文档是否有效？
  - 当某一文档调整后的影响，不同文档间如何保持一致性？
  - 大量的文档就一定有效吗？
  - 把用户隔离在开发过程之外，不容易满足用户需求。

瀑布模型带来的问题

- 对于大多数软件项目，客户看到软件前无法可靠地描述他们想要的是什么，而瀑布模型却要求客户明确需求，这就很难适应许多项目开始阶段必然存在的不确定性。
- 客户必须要有耐心，因为只有在项目接近尾声时，他们才能得到可执行的程序。对于系统中存在的重大缺陷，如果在可执行程序评审之前没有发现，将可能造成惨重损失。
- 瀑布模型在某些项目中容易导致“阻塞状态”。因为任务之间的依赖性，开发团队的一些成员要等待另一些成员工作完成才能进行下一步工作。

瀑布模型带来的问题

- **平均水平的项目开发过程中会有25%，大型项目有40%的需求变化**
- **需求变更导致的返工占总返工量的80%**
  - 瀑布方法需求中45%从未被使用，时间计划与实际相差4倍

瀑布模型

- 瀑布模型严格按计划执行，把用户隔离在开发阶段外面，用户看见产品要修改只能在后期，但修改又困难，在大型系统开发中已经很少使用。
- 瀑布模型很直观，符合惯性思维，容易理解，是其它模型的基础。
- 瀑布模型适用场合：
  - 软件项目较小；
  - 需求在项目开始之前已经被全面的了解；
  - 需求在开发中不太可能发生重大改变；
  - 外部环境的不可控因素很少。

瀑布模型优点与缺点

- 优点
  - 简单、易懂、易用、快速；
  - 项目划分为多个阶段，按阶段划分检查点，项目管理比较容易；
  - 每个阶段必须提供文档，而且要求每个阶段的所有产品必须进行正式、严格的技术审查，可操作性强。
- 缺点
  - 在开发早期，用户难以清楚地确定所有需求，需求的错误很难在开发后期纠正，因此难以快速响应用户需求变化。
  - 开发人员的工作几乎完全依赖规格说明文档，开发人员与用户之间缺乏有效的沟通，不容易满足客户需求。
  - 客户必须在项目接近尾声的时候才能得到可执行的程序，对系统中存在的重大缺陷，如果在评审之前没有被发现，将可能会造成重大损失。

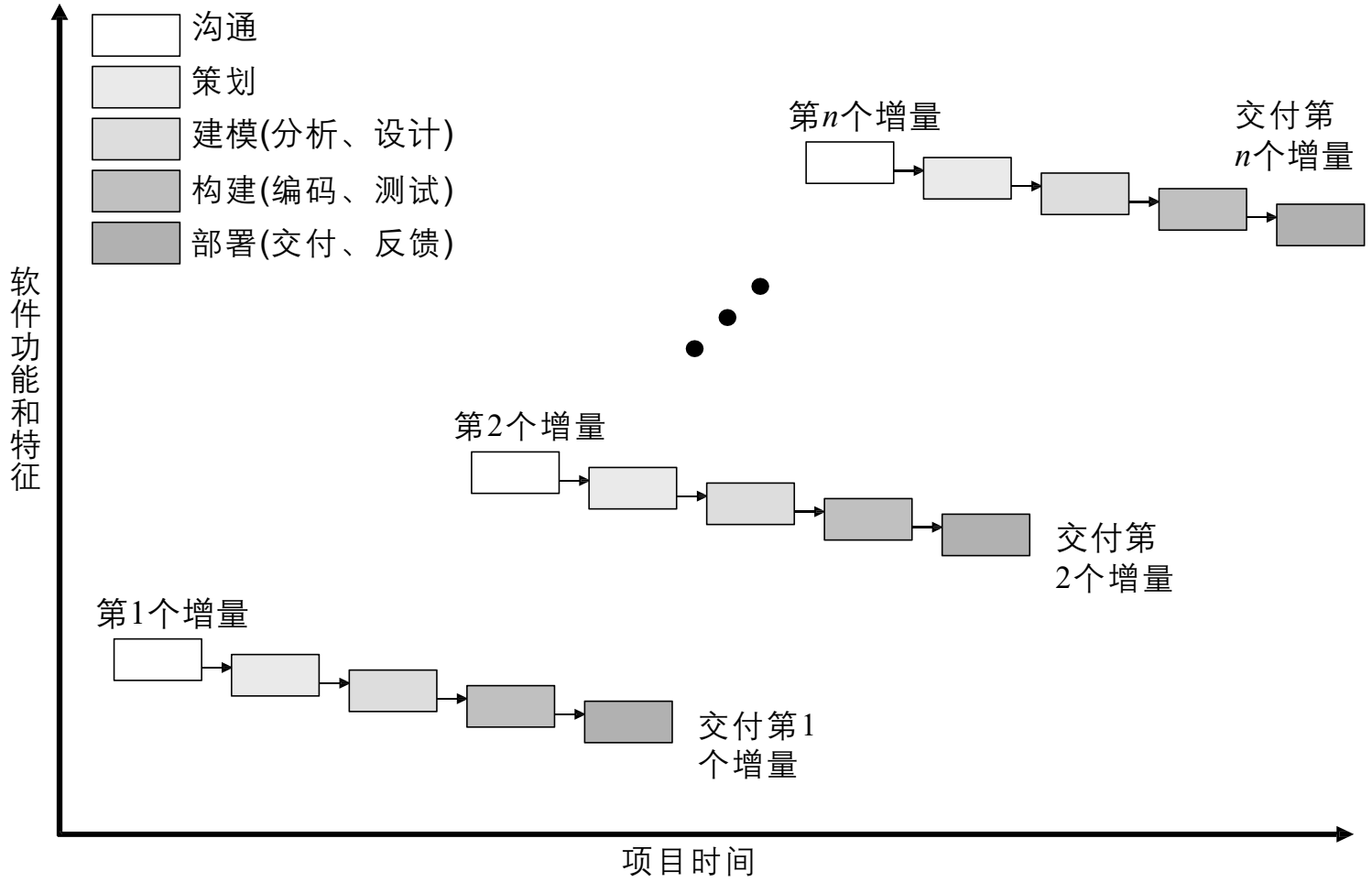
典型软件过程模型

- 瀑布模型
- 增量过程模型
  - 增量模型
  - 快速应用程序开发（RAD）
- 演化过程模型
  - 快速原型开发模型
  - 螺旋模型

增量过程模型

- **为什么需要增量过程模型？**
  - 在许多情况下，初始的软件需求有明确的定义，但是整个开发过程却不宜单纯运用线性模型。
  - 可能迫切需要为用户迅速提供一套功能有限的软件产品，然后在后续版本中再进行细化和扩展功能。

增量过程模型——增量模型


沟通
策划
建模(分析、设计)
构建(编码、测试)
部署(交付、反馈)
第n个增量
交付第
n个增量
第2个增量
交付第
2个增量
第1个增量
交付第1
个增量
软件功能和特征
项目时间

增量模型使用方法

- **使用方法：软件被作为一系列的增量来进行开发，每一个增量都提交一个可以操作的产品，可供用户评估。**
  - 第一个增量往往是核心产品：满足了基本的需求，但是缺少附加的特性；
  - 客户使用上一个增量的提交物并进行评价，制定下一个增量计划，说明需要增加的特性和功能；
  - 重复上述过程，直到最终产品产生为止。
- **本质：以迭代的方式运用瀑布模型**

增量模型的应用举例

- **举例：开发一个类似于Word的字处理软件**
  - 增量1：提供基本的文件管理、编辑和文档生成功能；
  - 增量2：提供高级的文档编辑功能；
  - 增量3：实现拼写和语法检查功能；
  - 增量4：完成高级的页面排版功能；
  - 增量5：。。。

增量模型优点和缺点

- **优点：**
  - 在时间要求较高的情况下交付产品：在各个阶段并不交付一个可运行的完整产品，而是交付满足客户需求的一个子集的可运行产品，**对客户起到“镇静剂”的作用**；
  - **人员分配灵活**：如果找不到足够的开发人员，可采用增量模型，早期的增量由少量人员实现，如果客户反响较好，则在下一个增量中投入更多的人力；
  - 逐步增加产品功能可以**使用户有较充裕的时间来学习和适应新产品**，避免全新软件可能带来的冲击；
  - 因为具有较高优先权的模块被首先交付，而后面的增量也不断被集成进来，这使得最重要的功能肯定接受了最多的测试，从而使得**项目总体性失败的风险比较低**。
- **缺点：**
  - 每个附加的增量并入现有的软件时，**必须不破坏原来已构造好的东西**。
  - 同时，**加入新增量时应简单、方便**——该类软件的体系结构应当是开放的；

增量模型的适用情况

- **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。**
- **在开发中，期望尽快提交其中的一些增量产品。**

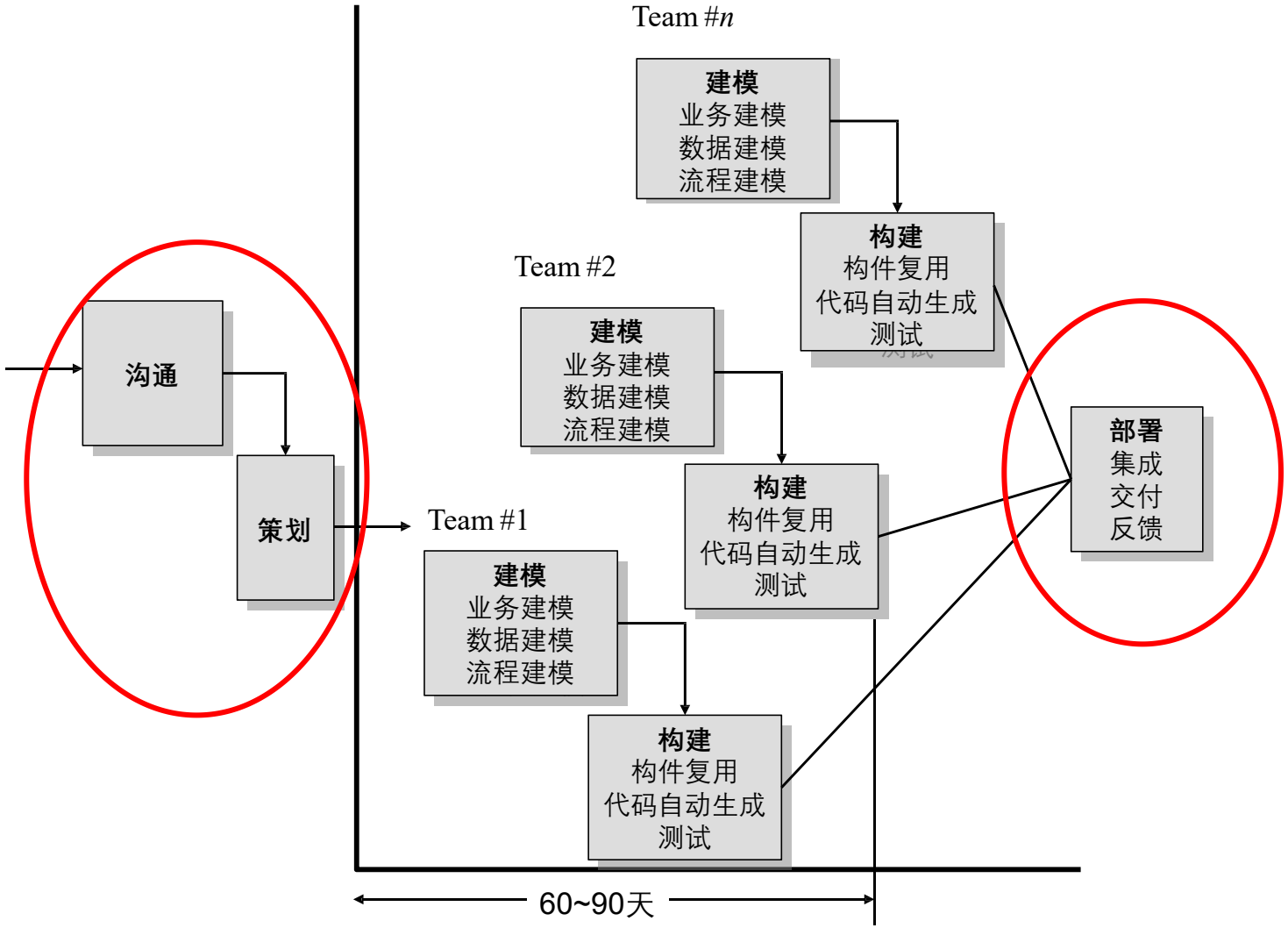
例如：

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。

典型软件过程模型

- 瀑布模型
- 增量过程模型
  - 增量模型
  - 快速应用程序开发（RAD）
- 演化过程模型
  - 快速原型开发模型
  - 螺旋模型

增量过程模型——快速应用开发（RAD）模型



案例：教务管理系统

- **教务管理系统**
  - 项目集中调研和规划，确定业务规范
  - 划分项目组，并行建模、构建、测试
    - Team1：学籍管理
    - Team2：成绩管理
    - Team3：选课管理
  - 集成测试及交付
  - 部署与反馈

RAD模型的使用方法

- 侧重于短开发周期(一般为60~90天)的增量过程模型，通过基于已有资源的构建方法实现快速开发。

- 本质：是瀑布模型的高速变体，并行运行瀑布模型。

RAD模型的优点和缺点

- 优点：
  - 提高软件交付速度
  - 充分利用企业已有资产进行项目开发
- 缺点：
  - 需求充分理解，系统被合理的模块化；
  - 需要大量的人力资源来创建多个相对独立的RAD团队；
  - 要求管理水平高，如果没有在短时间内为急速完成整个系统做好准备，RAD项目将会失败；
  - 如果系统需求是高性能，并且需要通过调整构件接口的方式来提高性能，不能采用**RAD模型**
  - 技术风险很高的情况下(采用很多新技术、软件需与其他已有软件建立集成、等等)，不宜采用RAD。

典型软件过程模型

- 瀑布模型
- 增量过程模型
  - 增量模型
  - 快速应用程序开发（RAD）
- 演化过程模型
  - 快速原型开发模型
  - 螺旋模型

演化过程模型

- 软件系统会随着时间的推移而发生变化，在开发过程中，需求经常发生变化，直接导致产品难以实现。
- 在上述情况下，需要一种专门应对不断演变的需求软件过程模型，即“演化过程模型”。
- 严格的交付时间使得开发团队不可能圆满完成整个软件产品，但是必须交付功能有限的版本以应对竞争或压力。
- 本质：循环、反复、不断调整当前系统以适应需求变化；
- 包括两种形态：
  - 原型开发模型
  - 螺旋模型

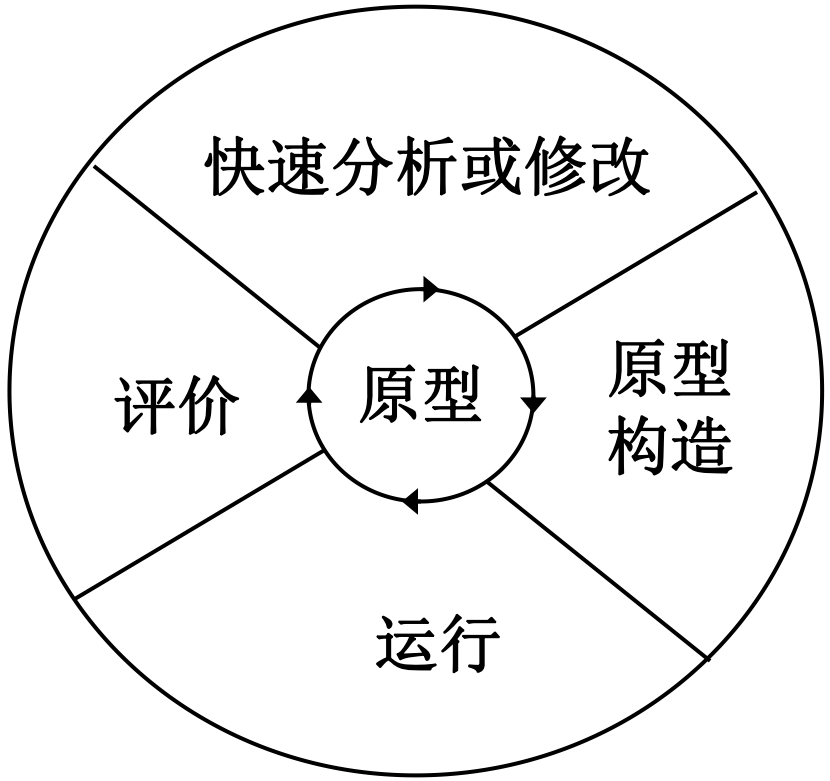
演化过程模型——快速原型开发

- 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

在原型系统不断调整以满足各种利益相关者需求的过程中，采用迭代技术，同时也使开发者逐步清楚用户的需求。



演化过程模型——快速原型开发

- **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。
- **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。
- **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。
- **原型修正**:对原型系统，要根据修改意见进行修正。
- **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

案例：教务管理系统

- **教务管理系统**
  - 第一次迭代：完成基本的学籍管理、选课和成绩管理功能；
    - 客户反馈基本满意，但是对大数据量运行速度慢效率，不需要学生自己维护学籍的功能等
  - 第二次迭代：加入权限控制，提高成绩统计和报表执行效率
    - 客户反馈：报表打印格式不符合要求
  - 第三次迭代：完善打印和权限控制功能；
  - 客户反馈：可以进行正式应用验证；

与增量模型的区别

- 增量模型：先构造一个核心功能，往里面加东西，每次是一个**可操作软件**，最后是产品的一个部分。
- 原型开发：得到基本需求后，简单分析就开始开发。用户不满意，**原型有可能抛弃**。原型是让用户来拿来进行评估和提意见的，可以是用户界面，或某一阶段的文档。根据用户的意见再完善。

原型开发的优点和缺点

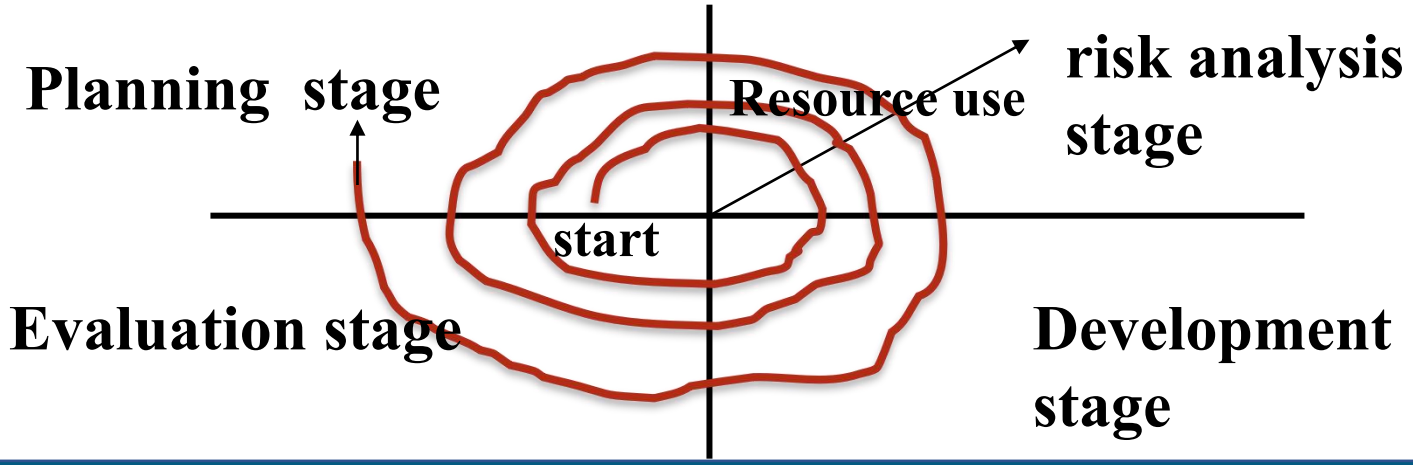
- **优点：**
  - 快速开发出可以演示的系统，方便与客户沟通；
  - 采用迭代技术能够使开发者逐步弄清客户的需求；
- **缺点：**
  - 为了尽快完成原型，开发者**没有考虑整体软件的质量和长期的可维护性，系统结构通常较差；**
  - **用户可能混淆原型系统与最终系统**，原型系统在完全满足用户需求之后可能会被直接交付给客户使用；

典型软件过程模型

- 瀑布模型
- 增量过程模型
  - 增量模型
  - 快速应用程序开发（RAD）
- 演化过程模型
  - 快速原型开发模型
  - 螺旋模型

演化过程模型—— 螺旋模型

- 该模型由Dr. Barry Boehm[Boehm 1988]提出。它是在瀑布模型和演化模型的基础上，加入两者所忽略的风险分析所建立的一种软件开发模型。
- 该模型将软件生存周期的活动分为四个可重复的阶段：规划、风险分析、开发和评估



演化过程模型—— 螺旋模型

其中：

- 评估和风险分析阶段都可作出一个决策：项目是否继续。
- 螺旋循环的次数指示了已消耗的资源；
- 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；
- 在早期螺旋循环中，为了为最终的实现给出一些指导性决策，经常使用原型构造；
- 设计和实现活动一般是在开发阶段进行；

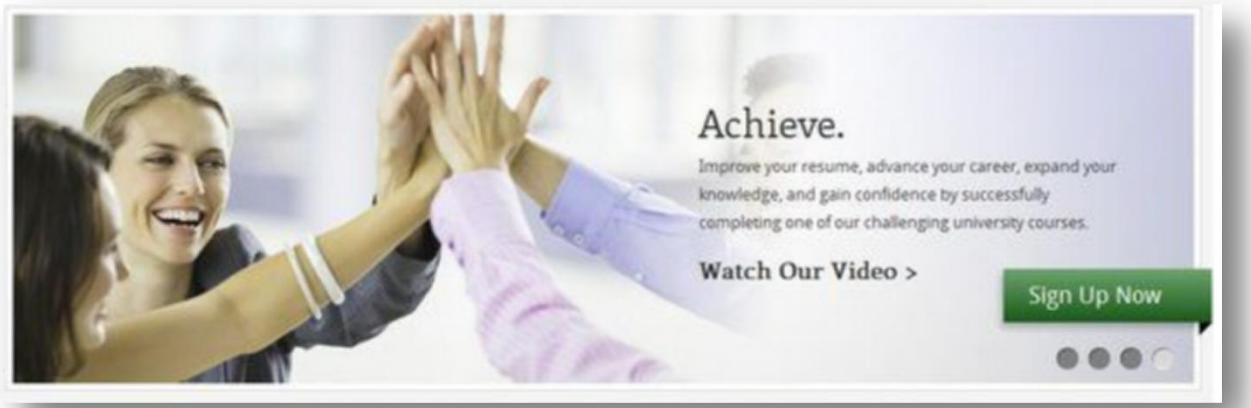
螺旋模型的优点和缺点

- **优点：结合了原型的迭代性质与瀑布模型的系统性和可控性，是一种风险驱动型的过程模型：**
  - 采用循环的方式逐步加深系统定义和实现的深度，同时更好的**理解、应对和降低风险**；
  - 确定一系列**里程碑**，确保各方都得到**可行的**系统解决方案；
  - 始终保持**可操作性**，直到软件生命周期的结束；
  - 由风险驱动，**支持现有软件的复用**。
- **缺点：**
  - 适用于大规模软件项目，特别是内部项目，**周期长、成本高**；
  - **软件开发人员应该擅长寻找可能的风险**，准确的分析风险，否则将会带来更大的风险；
  - 由于构建产品所需的周期数据不确定，给**项目管理带来困难**；
  - **演化速度不易把握**，演化速度太快，项目陷入混乱；演化速度太慢，影响生产率；
  - 为追求软件的高质量而**牺牲了开发速度、灵活性和可扩展性**；

案例分析

实例：网络公开课程网站

- 某公司准备开发一个大规模在线公开课程网站，支持学校将自己的课程录像、课件及参考资料等公布在网上，学生可以进行自主学习。
- 该系统将教育、娱乐和社交网络结合在一起，创造了一种新型的网络教育模式，对传统的高等教育模式带了很大的冲击。



案例分析

实例：网络公开课程网站

- 某公司准备开发一个大规模在线公开课程网站，支持学校将自己的课程录像、课件及参考资料等公布在网上，学生可以进行自主学习。
- 该系统将教育、娱乐和社交网络结合在一起，创造了一种新型的网络教育模式，对传统的高等教育模式带了很大的冲击。

实例分析：

- 系统需求会经常发生变化，业务模式存在不确定性
- 系统应该易于维护和修改
- 适合采用增量模型或演化模型

2.1 软件项目开发过程

- 软件生命周期、软件过程模型(软件生命周期模型)的概念及其关系
- 明白软件开发过程的典型阶段
- 典型软件过程模型

  - 瀑布模型
  - 增量过程模型
    - 增量模型
    - 快速应用程序开发（RAD）
  - 演化过程模型
    - 原型开发模型
    - 螺旋模型

- 对比分析几种典型的软件过程模型的异同

各种过程模型的比较

- **针对本次课程所讲授的软件过程模型，通过对比分析，找出这些过程模型之间的异同，并分析各自的优缺点；**
- **选择模型时考虑以下维度：**
  - 时间效率、成本、人力资源、开发质量、顾客满意度、需求扩展、需求变化、风险、与顾客交互程度、适用项目规模、适用deadline紧急程度、项目管理的方便程度、等等。