2.1 软件项目开发过程

- 软件生命周期，软件过程模型的概念及其关系
  - 软件生命周期
  - 软件过程模型
- 软件开发过程的典型阶段
  - 过程与过程方法
  - 典型阶段

    计划 -> 需求分析 -> 软件设计 -> 软件实现 -> 软件验证 -> 软件维护
    - 计划

      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。进行可行性研究，得出**可行性分析报告**
    - 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

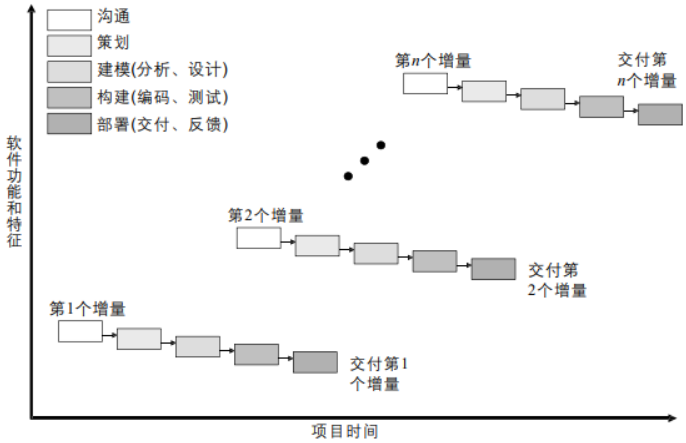
      系统投入使用后对其进行改进，以适应不断变化的需求。完全从头开发的系统很少，将软件系统的开发和维护看成是一个连续过程更有意义
  - 注意：各阶段的产物需要验证和确认

    开发的产品与用户的需求接近
- 典型软件过程模型
  - 瀑布模型
    - 瀑布模型的开发严格按照**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**
    - 特点
      - 需求最重要，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
    - 适用场景
      - 软甲项目较小

    - 需求在项目开始前已经被全面的了解
    - 需求咋你开发中不太可能发生重大改变
    - 外部环境的不可控因素较少
  - 优点
    - **简单、易懂、易用、快速**
    - 项目分为多个阶段，按阶段划分检查点，**项目管理容易**
    - 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
    - 开发早期，用户不确定需求，难以快速响应用户**需求变化**
    - 开发人员工作依赖规格说明文档，与用户缺乏沟通，**不容易满足客户需求**
    - 用户在项目尾声才可以得到可执行程序，系统中重大缺陷不易及时排查，**可能造成重大损失**
- 增量过程模型

  迫切需要为用户提供一套功能有限的软件产品，随后在后续版本中进行细化和扩展功能

  - 增量模型

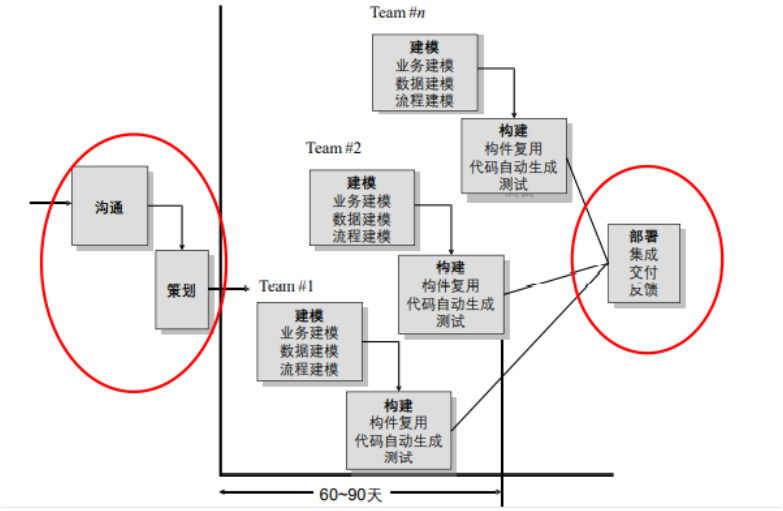
    

    - 使用方法

      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。

      - 第一个增量为核心产品，满足基本需求，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，指定下一个增量计划，说明需要增加的特性和功能
      - 重复上述过程，最终产品产生为止
    - 本质：以迭代的方式运用瀑布模型
    - 优点
      - 在时间要求较高的情况下交付产品，每次交付满足用户需求子集的可运行产品，**对用户起镇静剂的作用**
      - **人员分配灵活**，早期的增量由少量人员实现，若反响好，下一增量投入更多人力

        - 逐步增加功能**使用户有充裕时间学习和适应新产品**
        - 较高优先权模块首先交付，最重要功能接受了最多测试，**项目总体性失败的风险较低**
    - 缺点
        - 每个附加增量并入现有软件时，**必须不破坏原有已构造好的东西**
        - **加入新增量时应简单、方便**，该类软件体系结构应是开放的
    - 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。
- 快速应用程序开发（RAD）



    - 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

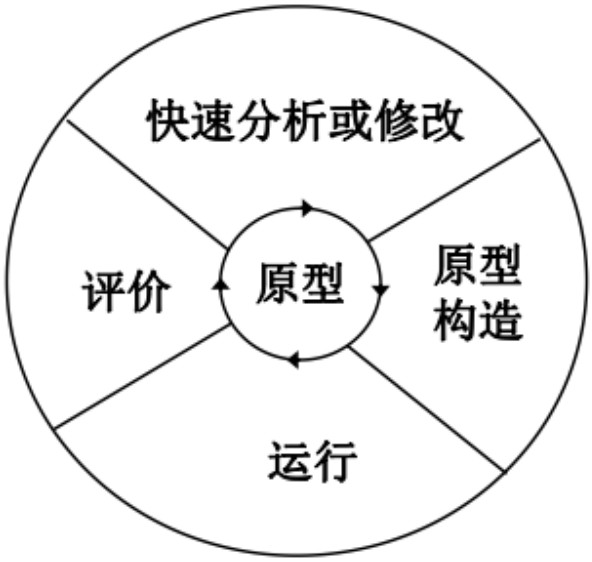
      是瀑布模型的高速变体，并行运行瀑布模型
    - 优点
        - 提高软件交付速度
        - 充分利用企业已有资产进行项目开发
    - 缺点
        - **需求充分理解，系统被合理的模块化**
        - **需要大量的人力资源**来创建多个相对独立的RAD团队
        - **要求管理水平高**，如果**没有短时间内为急速完成整个系统做好准备**，项目失败
        - **系统需求是高性能，并且需要通过调整构件接口的方式来提高性能，不能采用RAD模型**
        - **技术风险很高的情况**不宜采用RAD
- 演化过程模型

专门应对不断演变的需求软件过程模型

- 本质

  循环、反复、不断调整当前系统以适应需求变化

- 快速原型开发模型



  - 快速原型

    快速原型是快速建立起来的可以在计算机上运行的程序，其所能完成的功能是最终产品完成功能的一个子集

    - 作用
      - 获得用户的真正需求
      - 用于为一个项目中某些部分，确定技术、成本和进度的可能性
  - 过程
    - 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
  - 与增量模型的区别
    - 增量模型

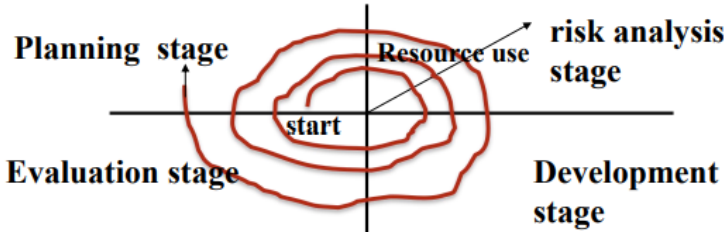
      构造一个核心功能，添加功能，每次得到可操作性软件，最后是产品的一部分

- 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。

- 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
  - 为快速完成原型，**开发者没有考虑整体软件的质量和长期的可维护性，系统结构较差**
  - **用户可能混淆原型系统与最终系统**，原型该系统在完全满足用户需求之后可能被直接交付客户使用

- 螺旋模型



> 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

- 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**
  - 采用循环方式逐步加深系统定义与实现深度，**更好理解、应对降低风险**
  - 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**
  - 由风险驱动，**支持现有软件的复用**
- 缺点
  - 适用于大规模软甲你项目，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**
  - 构建产品所需周期数据不确定，给**项目管理带来困难**
  - **演化速度不易把握**
  - 为追求软件高质量牺牲了**开发速度、灵活性和可扩展性**

- 软件过程模型比较
  - 从以下角度考虑

```
-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。
```