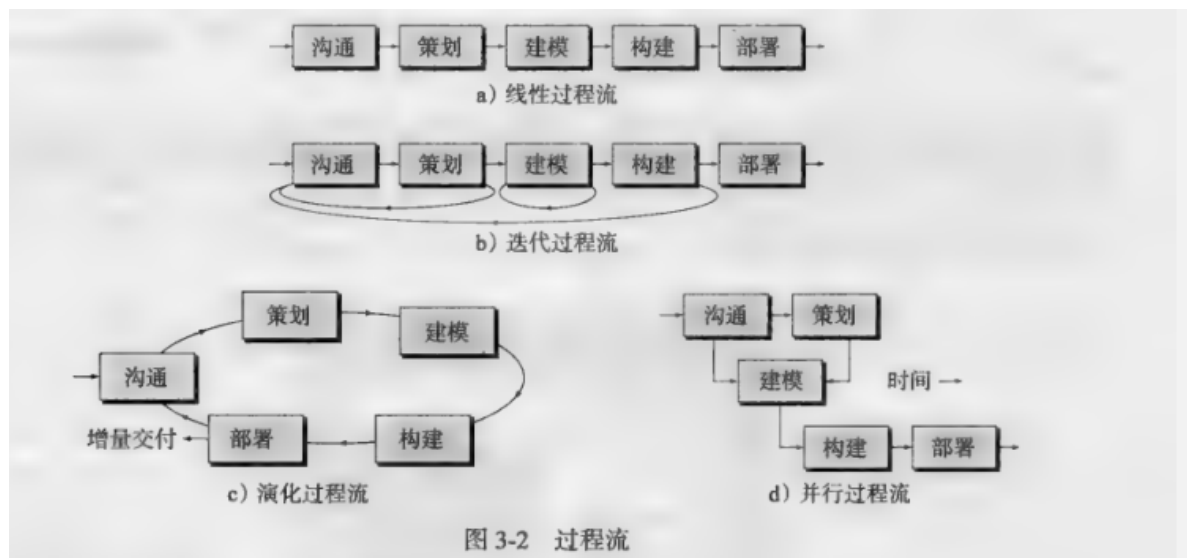


## Ch3 软件过程框架

- 过程流

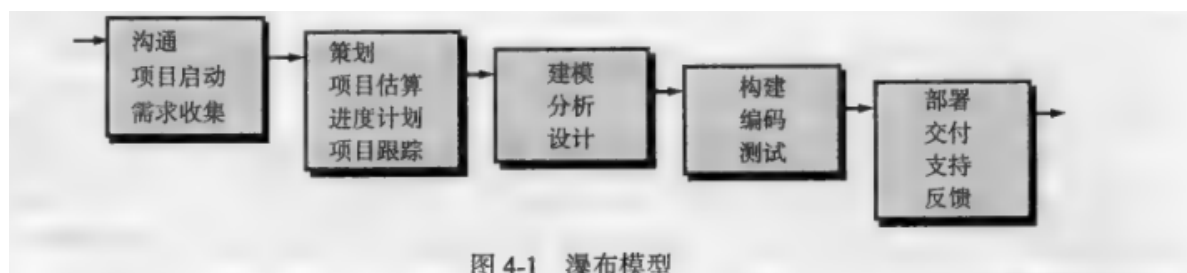


- ), 沟通活动可能会包含 6 个不同的动作(具体参见第 8 章): 起始、需求获取、需求细化、协商、规格说明和确认。
- 过程评估标准
  - 用于过程改进的 CMMIS 标准评估方法( Standard CMMI Assessment Method for Process Improvement, SCAMPI)
  - 于组织内部过程改进的 CMM 评估 (CMM-Based Appraisal for Internal Process Improvement, CBA IPD)
  - SPICE (ISO/IEC 15504)
  - 软件ISO 9001:2000

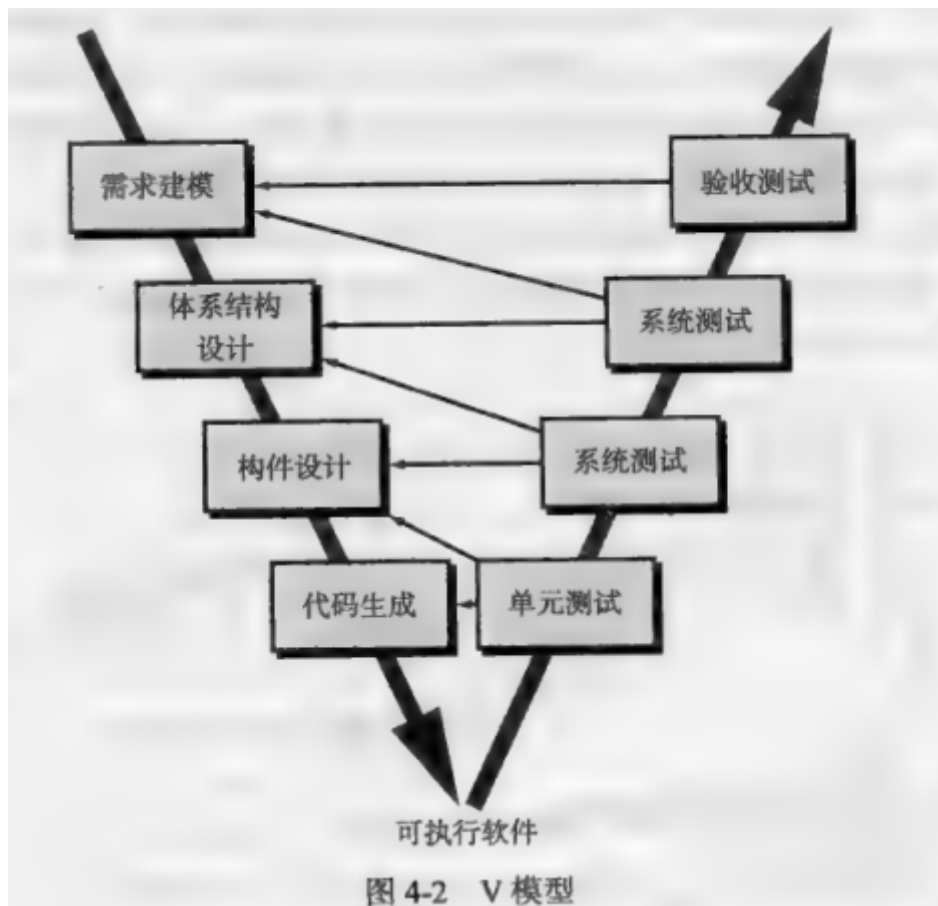
## 过程模型

### 惯用过程模型

- 瀑布模型:瀑布模型 (waterfall model) 又称为经典生命周期 ( classic life cycle),
  - 生在需要对一个已经存在的系统进行明确定义的适应性调整或是增强的时候
  - 需求必须是准确定义和相对稳定的



- 瀑布模型的变体:V 模型



- 增量过程模型: A good approach when a working core product is required quickly.
- 演化过程模型:
  - Are iterative in nature.
  - B. Can easily accommodate product requirements changes.
  - C. Do not generally produce throwaway systems.
  - 原型开发模型: A useful approach when a customer cannot define requirements clearly.
  - 客户定义了软件的一些基本任务，但是没有详细定义功能和特性需求。另一种情况下，开发人员可能对算法的效率、操作系统的适用性和人机交互的形式等情况并没有把握。
  - 螺旋模型: 结合了原型的迭代性质和瀑布模型的可控性和系统性特点，它具有快速开发越来越完善的软件版本的潜力；Includes project risks evaluation during each iteration.
- 并发模型: 并发工程；发建模定义了一系列事件，这些事件将触发软件工程活动、动作或者任务的状态转换。

## 专用过程模型

- 基于构件的开发
  - 基于构件的开发模型 (component-based development model) 具有许多螺旋模型的特点。它本质上是演化模型，需要以迭代方式构建软件。不同之处在于，基于构件的开发模型采用预先打包的软件构件来开发应用系统。
- 形式化方法模型 formal methods model

- A. Define the specification for computer-based systems.
- B. Develop defect free computer-based systems.
- C. Verify the correctness of computer-based systems.
- 面向方面的软件开发
- 

## 统一过程

- 起始阶段 (inception phase)
- Elaboration phase 细化阶段
- Construction phase 构建阶段
- Transition phase 转化阶段
- Production phase 生产阶段

## 个人软件过程 (PSP)

## 团队软件过程(TSP)

TSP 定义了以下框架活动: 项目启动、高层设计、实现、集成和测试以及后

传统软件过程模型已经使用了多年,力图给软件开发带来秩序和结构。每一个模型都建议了一种不同的过程流,但所有模型都实现同样的一组通用框架活动:沟通、策划、建模、构建和部署。

像瀑布模型和 V 模型这类顺序过程模型是最经典的软件工程模型,顺序过程模型建议采用线性过程流,这在软件世界里通常与当代的软件开发现实情况不符(例如持续的变更、演化的系统、紧迫的开发时间)。但线性过程模型确实适用于需求定义清楚且稳定的软件开发。

增量过程模型采用迭代的方式工作,能够快速生成一个软件版本。演化过程模型认识到大多数软件工程项目的迭代、递增特性,其设计的目的是为了适应变更。演化模型,例如原型开发及螺旋模型,会快速地产生增量的工作产品(或是软件的工作版本)。这些模型可以应用于所有的软件工程活动——从概念开发到长期的软件维护。

并发过程模型为软件团队提供了过程模型中的重叠和并发元素的描述方法。专用模型主要包括基于构件的模型,强调软件构件的重用和组装;形式化方法模型提倡采用数学方法进行软件开发与验证;面向方面的模型的目的是解决跨越整个软件体系结构的横切关注问题。统一过程模型是一种“用例驱动、以体系结构为核心、迭代及增量”的软件过程框架,由 UML 方法和工具支持。

软件过程的个人模型和团队模型都强调成功软件过程的关键因素:测量、策划和自我管理。

## 敏捷开发

- 敏捷团队是能够适当响应变更的灵活团队

## 极限编程 Extreme Programming (XP)

planning, design, coding, testing

## 工业极限编程

由 XP 的最低限要求、以客户为中心和测试驱动精神组成

## Scrum game

过程由“需求、分析、设计、演化和交付”等框架性活动组成

1) 上次 Scrum 例会后做了什么?  
2) 遇到了什么困难?  
3) 下次例会前计划做些什么?

## 动态系统开发方法 ( Dynamic System Development Method, DSDM)

## 敏捷模型

作为首选的方法来表示分析和设计模型。

## 测试策略

---

- 传统软件：集成测试；单元测试
- 面向对象：同上；（集成测试：基于线程的测试，基于使用的测试；）
- Web：内容，数据库，用户接口，可用性，构件级测试，导航栏测试，配置测试，安全测试，性能测试
- 移动：确认测试；系统测试；恢复测试；安全测试；压力测试；性能测试；部署测试

## 白盒测试

- 基本路径测试
  - 环复杂性计算

## 黑盒测试

- 基于图的测试
- 等价类划分
- 边界值分析
- 正交数组测试
- 

## 实时系统测试

## RMMM

对于风险管理，更好的是主动风险策略。主动 (proactive) 风险策略早在技术工作开始之前就已经启动了。识别出潜在的风险，评估它们发生的概率及产生的影响，并按其重要性进行排序

项目风险，技术风险，商业风险

风险管理策略可以包含在软件项目计划中，也可以将风险管理步骤组织成一个独立的风险缓解、监测和管理计划

