

Chapter 2

■ Process Models

Slide Set to accompany

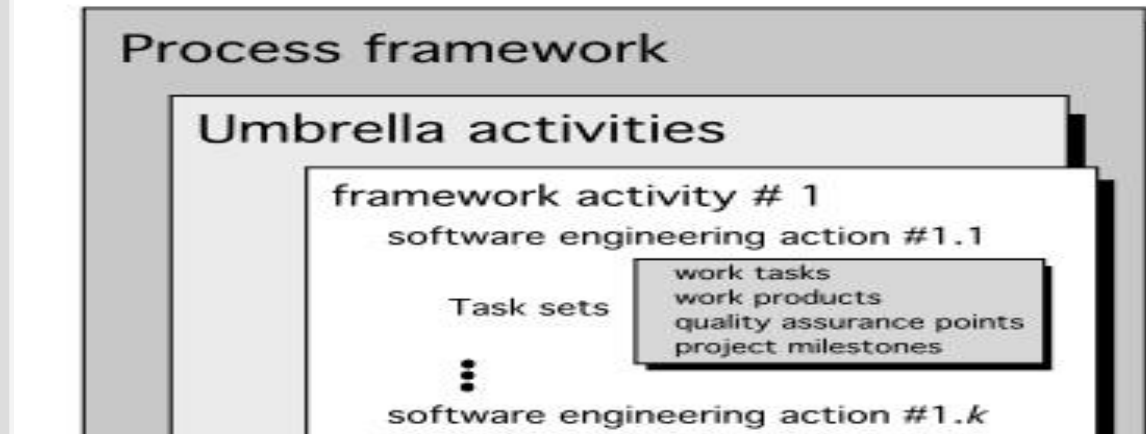
Software Engineering: A Practitioner's Approach, 7/e

by Roger S. Pressman

A Generic Process Model

- 软件过程是软件产品构建时所执行的一系列活动、动作和任务的集合。
- 过程模型定义
 - Define process flow
 - Define actions in each activity
 - Define task sets in each action

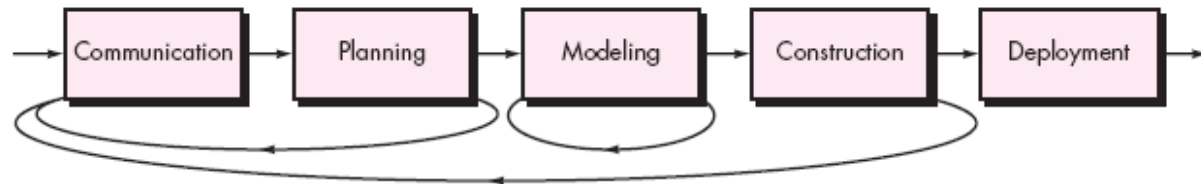
Software process



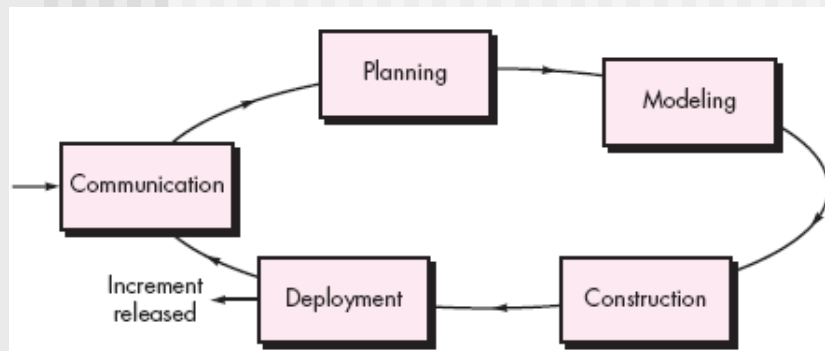
Process Flow



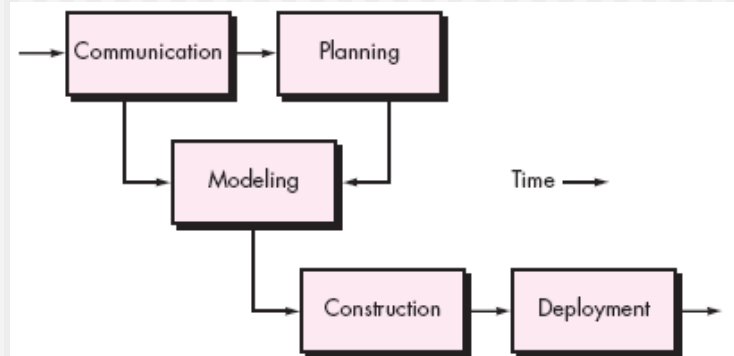
(a) Linear process flow



(b) Iterative process flow



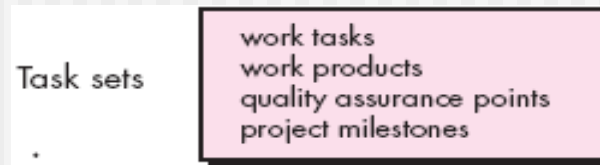
(c) Evolutionary process flow



(d) Parallel process flow

Identify TaskSet in Action

- A task set defines the actual work to be done to accomplish the objectives of an action.



- (1) A list of the task to be accomplished
- (2) A list of the work products

业务需求分析	准备调研提纲、问卷	咨询实施顾问	《需求调研提纲》	《一般业务调研提纲》 《分产品的调研提纲》 《行业调研提纲》
	制定调研计划	**项目经理	《需求调研计划》	《需求调研计划》
	<u>详细业务需求调研</u>	**项目经理/ 咨询实施顾问		
	需求和产品匹配分析	**项目经理		
	需求分析报告	**项目经理	《需求分析报告》	《需求分析报告》

Identify Task Set in Action

(3) A list of the quality assurance filters

准备测试环境(硬件/软件)	技术顾问/咨询实施顾问		
编写测试案例	咨询实施顾问	《测试案例清单》	← 质量检查点
准备测试数据	咨询实施顾问		
制定测试计划	**项目经理	《测试计划》	
测试实施	技术顾问/咨询实施顾问		
测试问题的处理	技术顾问/咨询实施顾问	《测试问题记录清单》	
测试总结和确认	**项目经理	《测试报告》	← 里程碑

(4) A list of the project milestones

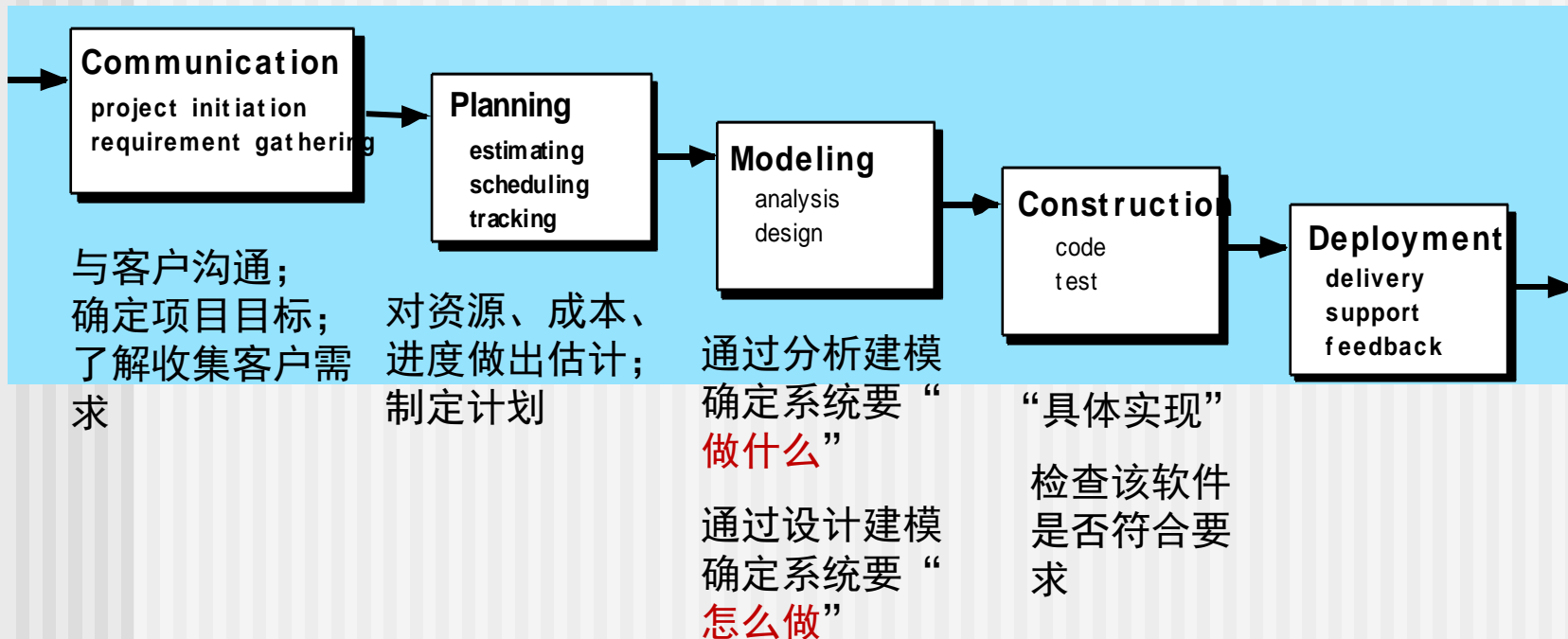
里程碑是项目开发过程中某个重要的时间节点，识别标志性工作的完成情况。

如：需求分析结束，就应该设置一个里程碑，对应标志性工作PRD的完成。

常用过程模型

- All software projects shall have a life cycle model which includes :
 - ◆ communication
 - ◆ planning
 - ◆ modeling
 - ◆ construction
 - ◆ deployment.

The Waterfall Model



- 强调了每一阶段活动的**严格顺序**。
- **推迟实现**。
- **质量保证**观点：每个阶段均提供**文档（里程碑）**，且需要通过**技术评审**，才能进行下一阶段的工作。

The Waterfall Model

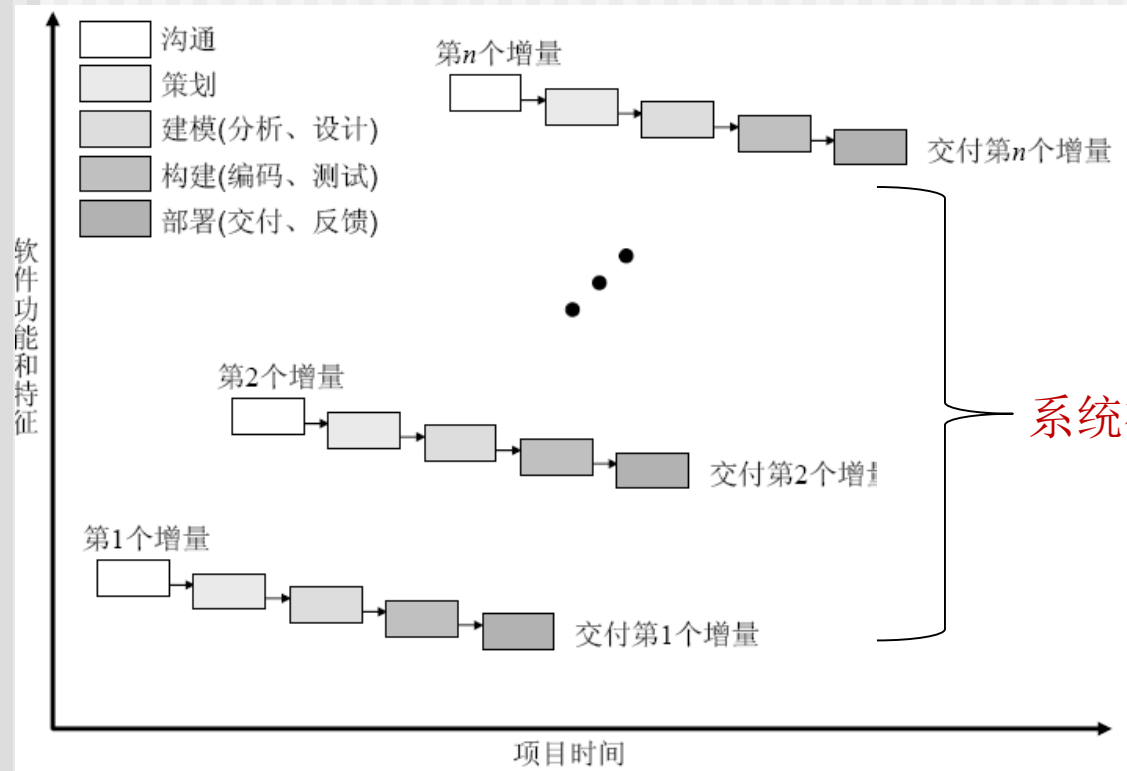
The problems :

- ✓ Real projects rarely follow the sequential flow.
- ✓ It is often difficult for the customer to state all requirements explicitly.
- ✓ The customer must have patience. A working version of the program will not be available until late in the project time-span.
- ✓ A major error, if undetected until the working program is reviewed, can be disastrous.

适合于:

- ✓ 需求稳定的项目
- ✓ 开发团队对该应用领域非常熟悉

The Incremental Model



- ✓ The incremental model combines elements of **linear** and **parallel** process flows.
- ✓ Each linear sequence produces a deliverable “increment” of the software.

The Incremental Model

- When an incremental model is used, the **first** increment is often **a core product**.
- The core product is used by the customer.
- A plan is developed for the next increment. The plan addresses the modification of the core product.

优势:

- 可提高对**用户需求**或**市场需求**的响应。
- 在**团队成员不足**以支持项目完整开发时尤其适用。
- 通过将不确定、不完全掌握相关技术的功能放在后续增量开发，可减轻技术风险。

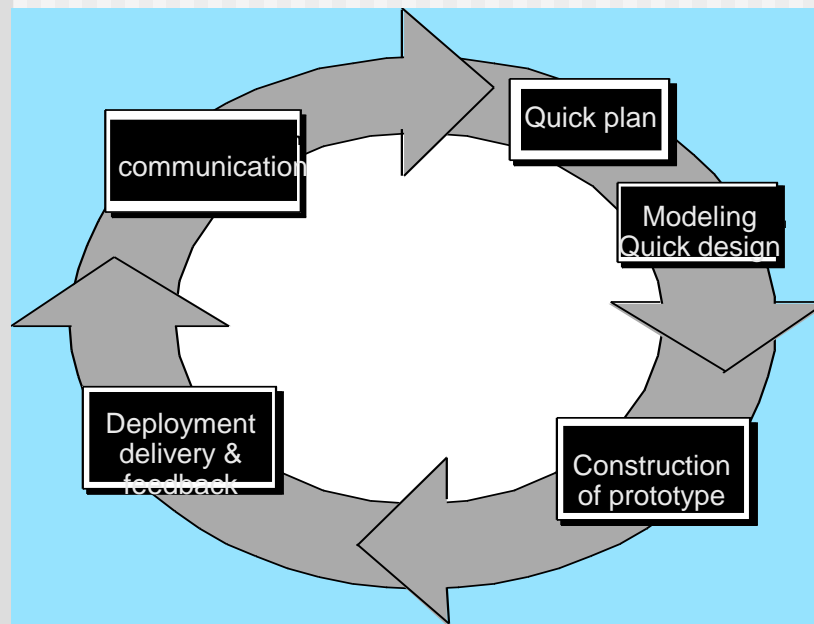
Apply Incremental Model

例如：通过增量模型，开发一文档编辑系统

- 增量1：提供基本的文档生成、编辑、存储功能；
- 增量2：提供更高级的编辑功能，如图表编辑等；
- 增量3：提供语法检查功能；

Evolutionary Models: Prototyping

- 演化模型的本质：通过迭代中一系列活动的重复应用，使软件持续改进。



- 让“系统”快速可见
- 减少分析、设计过程中的不确定性

(1) The prototype serves as a mechanism for identifying software requirements. 抛弃型

(2) The prototype can also slowly evolves into the actual system 演化型

Apply Prototyping(1)

例：应用抛弃型原型法，开发某库存系统

采购入库

入库单号

入库类型

采购入库

仓库名称

请选择

采购单号

入库时间

新增

编辑

删除

审核

打印

入库单号	仓库名称	制单人	入库类型	入库时间	审核状态	审核人	审核时间	入库说明
<input type="checkbox"/> RK-1-2-3	福量仓库	祥子	采购入库	2017-12-13	待审核	XXXXXX	XXXXX	XXXXXXXX

新增入库单

*入库编号

RK201712081418515772395

制单人员

XXX

*选择仓库

华北仓

入库时间

2017/12/08 14:18:51

采购订单

不选采购订单也可以入库

备注

选择商品

商品导入

删除行

商品编码	商品名称	规格名称	条形码	入库数量	采购数量	已入库	商品成本	小计
				5				

Apply Prototyping(2)

例：应用演化型原型法，开发“教务管理系统”

第一次原型迭代：完成基本的学籍管理、课程管理、成绩管理功能。
客户反馈：基本满意，但数据量变大时执行效率下降。

第二次原型迭代：修改细节，提高成绩统计和报表执行效率。
客户反馈：需要改进学生权限，报表格式不符合要求。

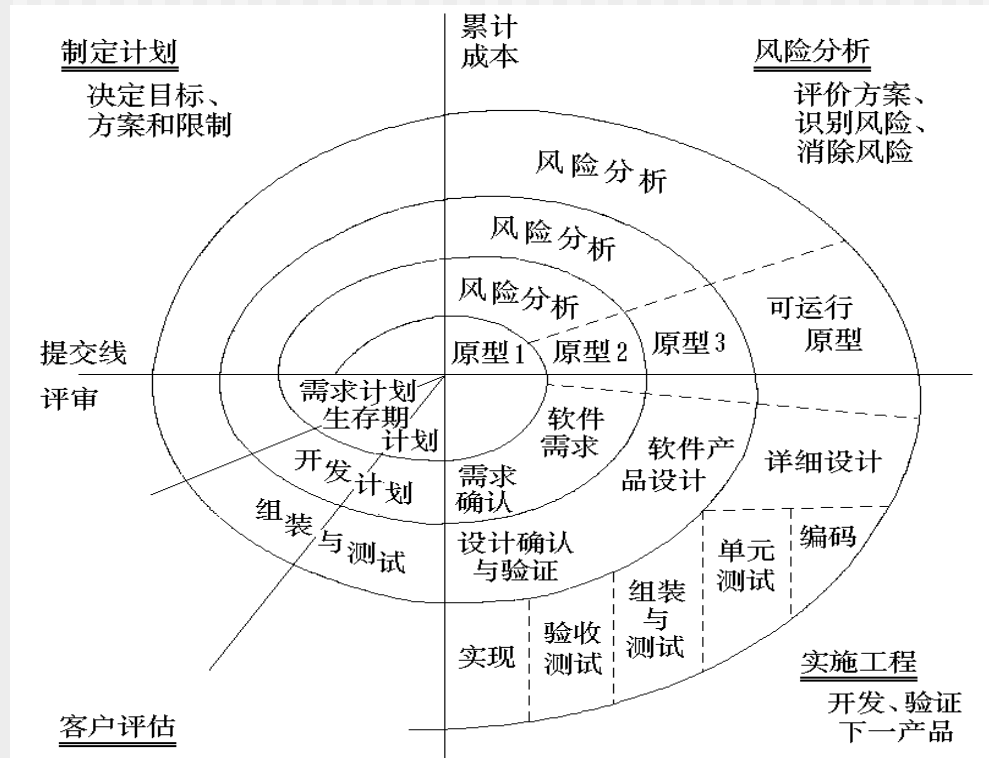
第三次原型迭代：完善报表打印和权限控制功能。
客户反馈：符合要求，可进行系统测试。

Evolutionary Models: Prototyping

The problems:

- ❑ 更高的开发代价
- ❑ 用户可能混淆原型系统和最终系统
- ❑ 开发人员为快速构建原型，需要在实现上作出妥协（例如：只关注局部而非整体结构，选择简单的算法）
- ❑ 需要快速原型开发工具和客户的深度参与，否则原型开发容易失败

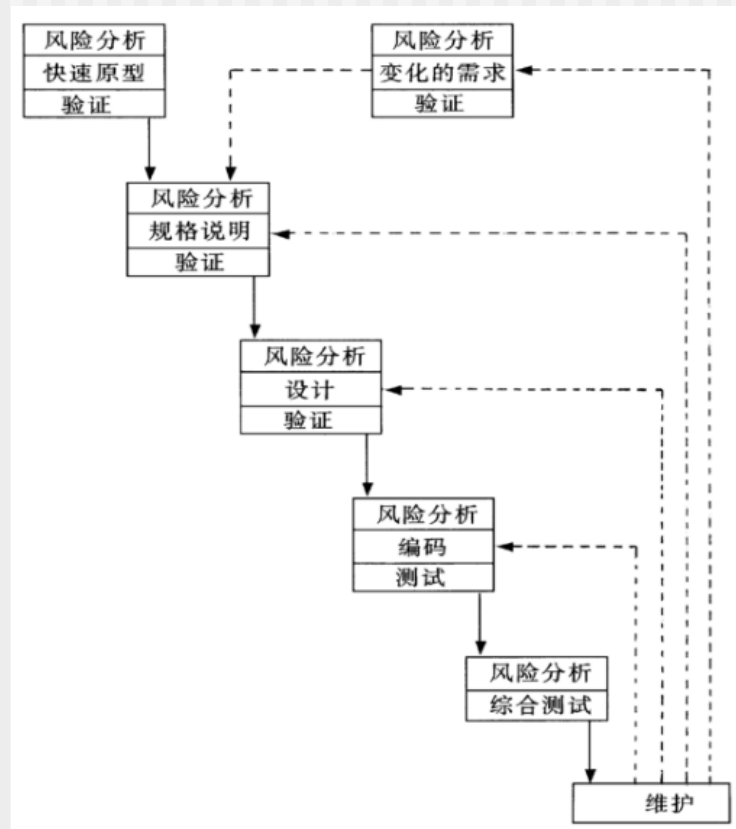
Evolutionary Models: Spiral



- 结合了原型模型的迭代特征和瀑布模型的系统性和可控性，是一种风险驱动的过程模型

Evolutionary Models: The Spiral

- 早期的内圈，可能仅仅产生产品的规格说明或一个简单原型；
- 后期每一圈，逐渐演化产生软件产品的初始版本，更完善的新版本.....



Evolutionary Models: The Spiral

特点:

- The spiral model is a realistic approach to the development of **large-scale systems** and software.
- The spiral model uses **prototyping as a risk reduction mechanism**
- It maintains the **systematic stepwise approach** but incorporates it into an iterative framework （**逐步加深**系统定义和实现的**深度**）

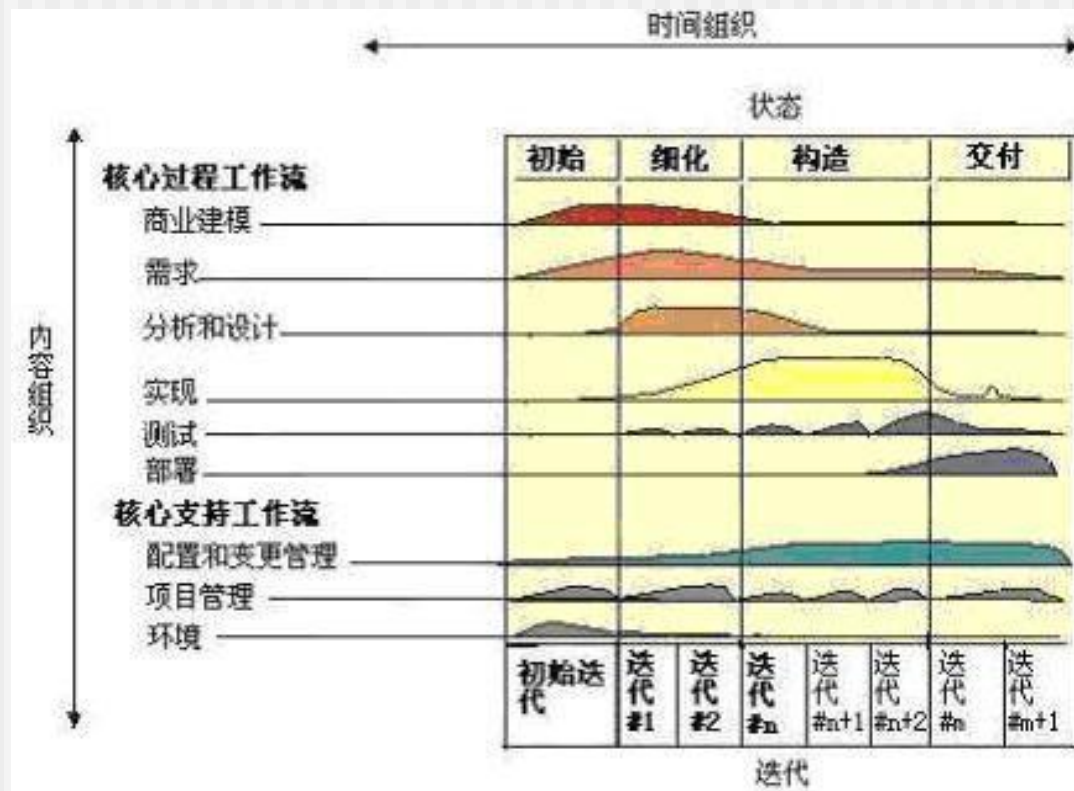
适合于:

- 大规模软件项目
- 高风险项目

The Unified Process (UP)

UP将项目生命周期分为四个阶段：

初始 细化 构造 交付



The Unified Process (UP)

- ✓ 每一个阶段由若干次迭代组成，只不过每一次迭代开发的侧重点不同（每个阶段集中力量做正确的事）。
- ✓ 一个迭代构成一个完整的开发循环，产生一个可执行的产品版本，是最终产品的一个子集。

Depict of UP Phases(1)

在UP初始阶段的迭代中，项目组必须：

- ✓ 确定开发目标与范围
- ✓ 评估潜在的技术、商业风险
- ✓ 制定迭代实施计划
- ✓ 识别系统的关键用例(10%左右)
- ✓ 选择合适的软件架构
 - 例如编译软件适合的架构为管道-过滤器风格
- ✓ 如果需要构造原型，可以进行设计和实现

Depict of UP Phases(2)

到了细化阶段的迭代，需要：

- ✓ 识别大部分用例？（至少80%）
- ✓ 通过多次迭代，每次选择当前最关注的用例进行体系结构设计，细化为对应的组件模型、实现模型、部署模型等。
- ✓ 核心是需求、分析和设计。
- ✓ 实现重要的部分用例，验证体系结构的稳定性。

Depict of UP Phases(3)

- ✓ 进入构建阶段后，核心是所有功能的实现、测试，为应用部署做好准备
- ✓ 最后到了交付阶段，核心是部署、系统上线、运行和维护。

UP裁剪

- UP是一个通用的过程模板，非常庞大。对不同的开发机构和项目，使用UP时需要**裁剪**，如：
 - ✓ 确定本项目需要哪些 workflow。RUP的9个核心 workflow 并不总是需要的，可以取舍。
 - ✓ 确定每个 workflow 需要哪些制品。
 - ✓ 确定每个阶段内的迭代计划。规划RUP的4个阶段中每次迭代开发的内容。

课堂练习

- ✓ 某公司开发一个通用CAD软件产品, 产品需求是逐步完善的, 某些需求在一定 范围内是明确的, 某些需求需要进一步细化, 但是迫于市场竞争的压力产品需要尽快上市,
- ✓ 问: 应该采用什么过程模型合适?
- ✓ 项目举例: 某公司需要给火车站开发一个交互式火车车次查询系统, 这是火车站首次使用该 系统, 因此需求不是很明确。
- ✓ 问: 应该采用什么过程模型合适?