

# 第二章 过程模型

王美红



# 提出问题.....

- 在开发产品或构造系统时，由谁指导我们及时交付高质量的产品？

所遵循的线路图就称为：

软件过程

# 主要内容

- 过程框架
- CMMI
- 过程模式
- 惯用过程模型
- 专用过程模型

## 2.1 过程框架

- 将整个软件过程再进一步细分为各个相对独立的功能块，即过程框架。（以工作开展的时间为线索）

## 过程框架

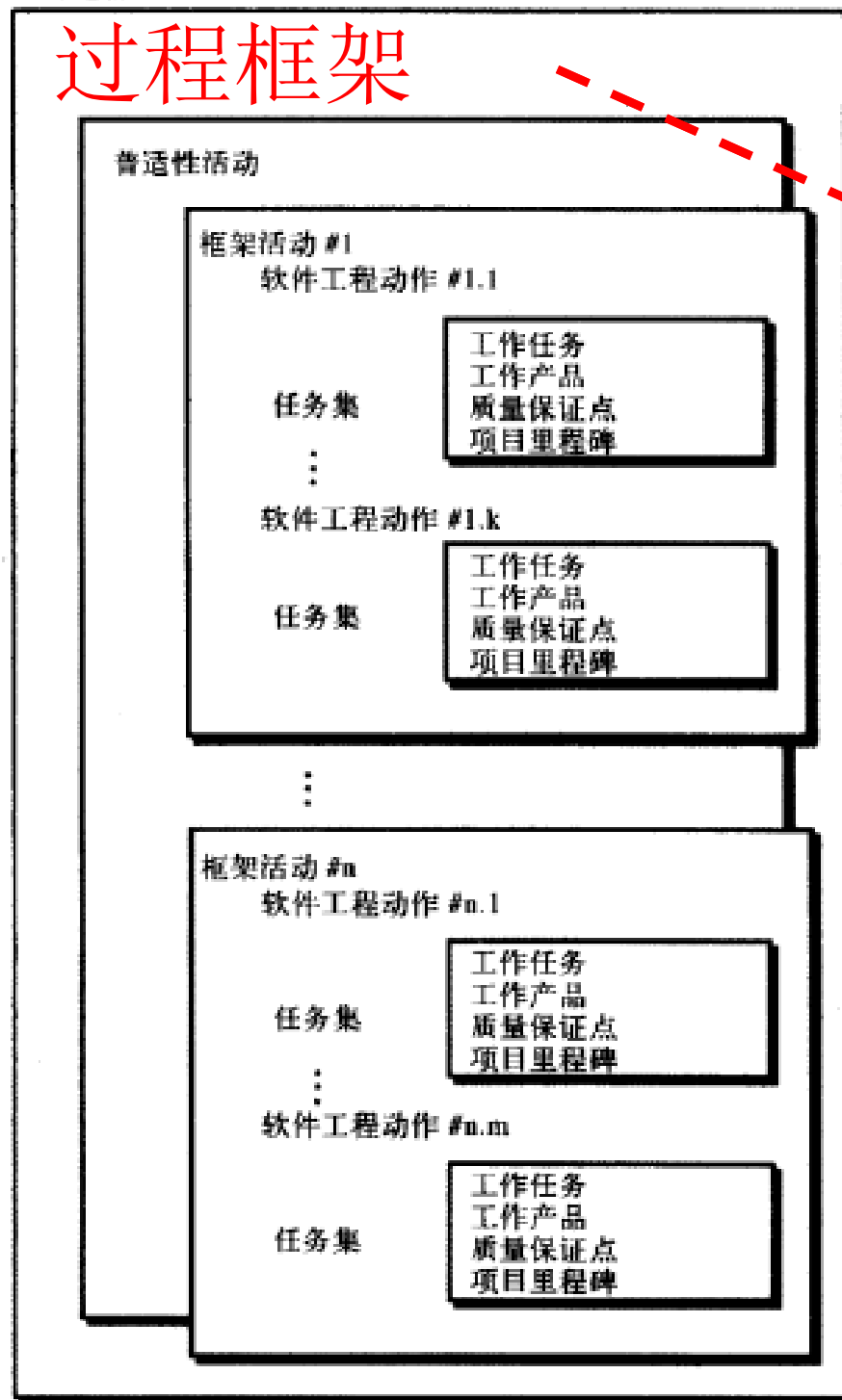
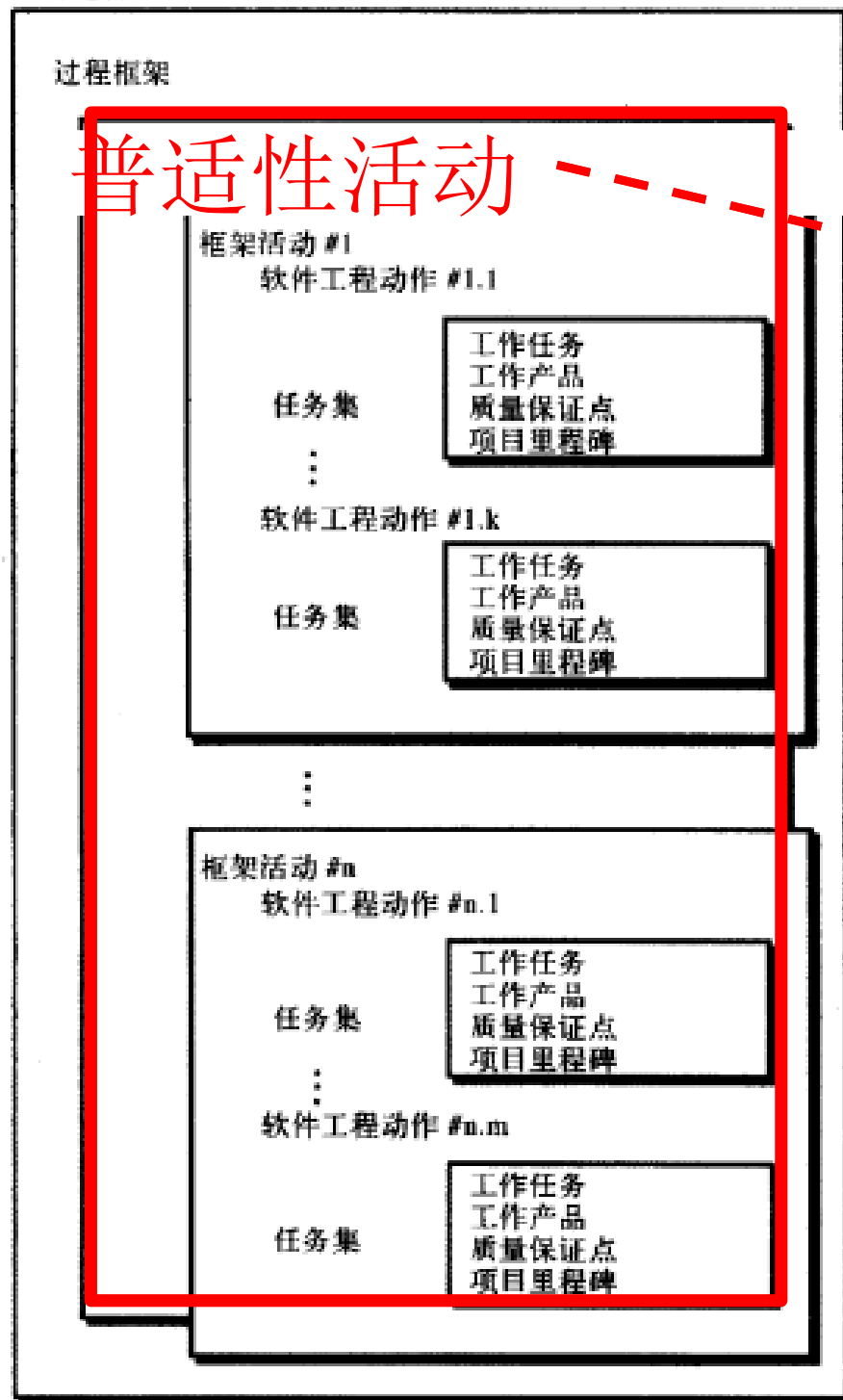
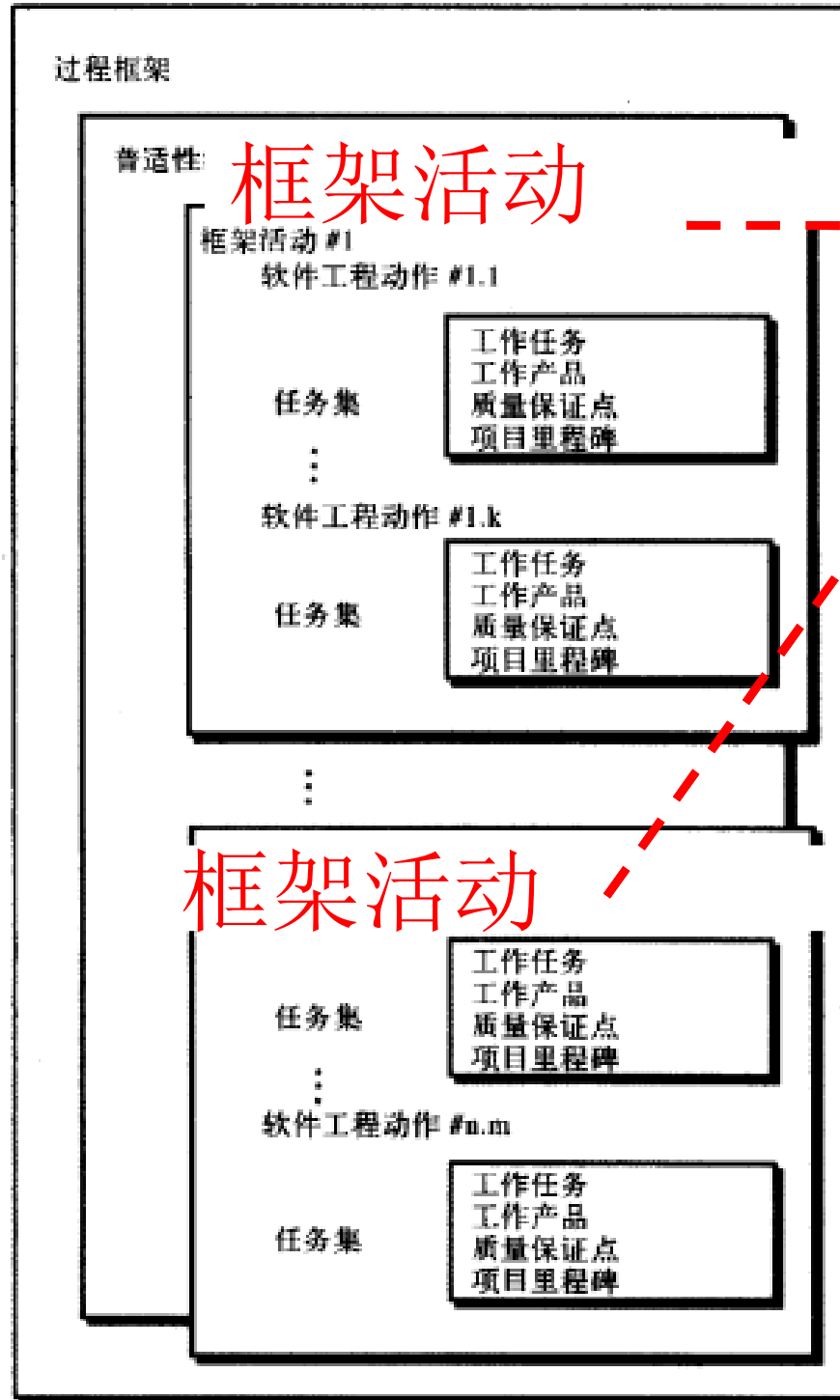
定义了若干个  
框架活动图  
2-1  
软件过程  
框架

图  
2-1  
软件过程  
框架



适用于任何一个  
框架活动

图  
2-1  
软件过程  
框架



五个最基本的  
框架活动: 沟  
通、策划、建  
模、构建和部  
署

# 五个最基本的框架活动

- 沟通：与客户之间的交流与协作
- 策划：为后续的软件工程工作制定计划
- 建模：包括分析和设计
- 构建：编码和测试
- 部署：软件交付用户，用户对其进行评估并反馈意见



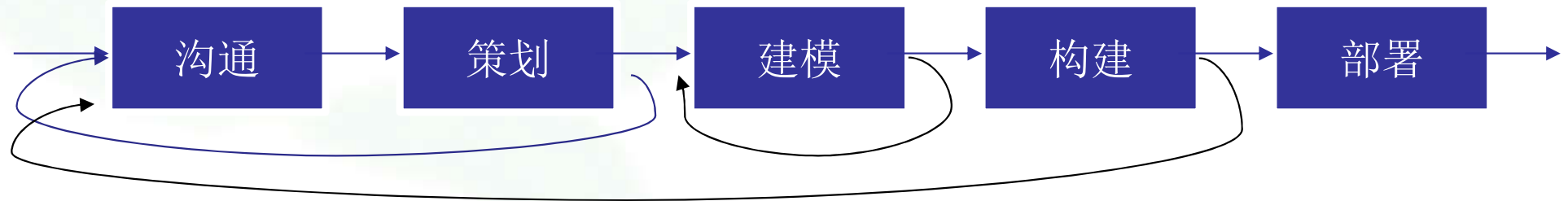
# 过程流

从沟通到部署线性执行



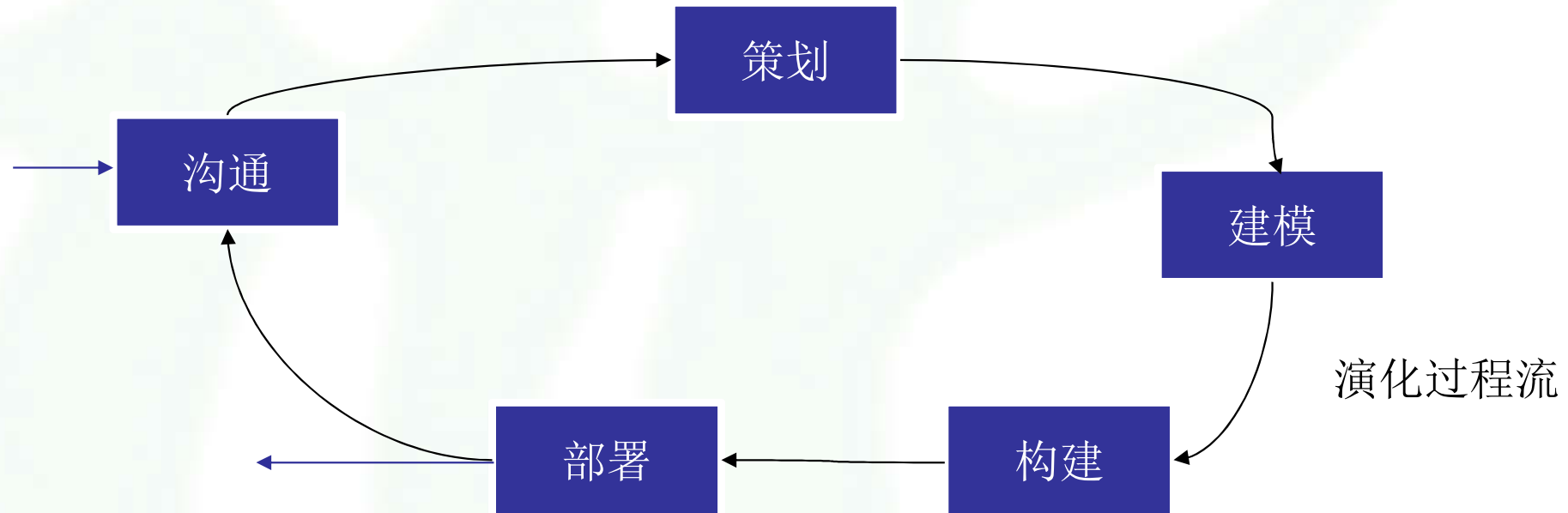
线性过程流

在执行下一个活动前重复执行之前的一个或多个活动



迭代过程流

采用循环的方式执行各个活动，每次循环都能产生更为完善的软件版本



将一个或是多个其它活动并行执行（如，软件一个方面的建模可以同软件另一个方面的建模活动并行执行）

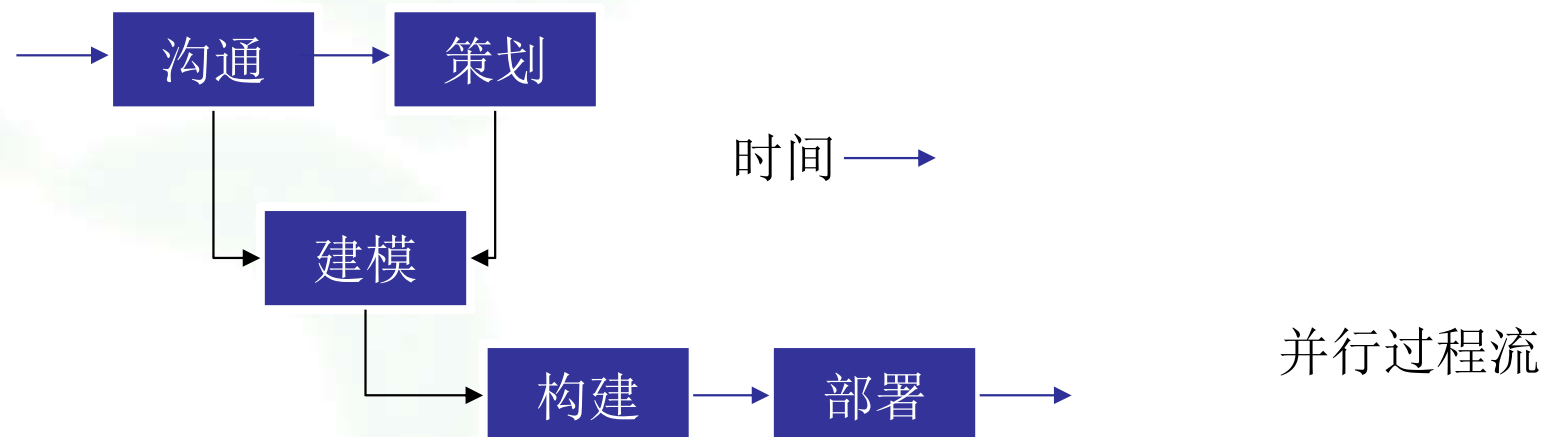
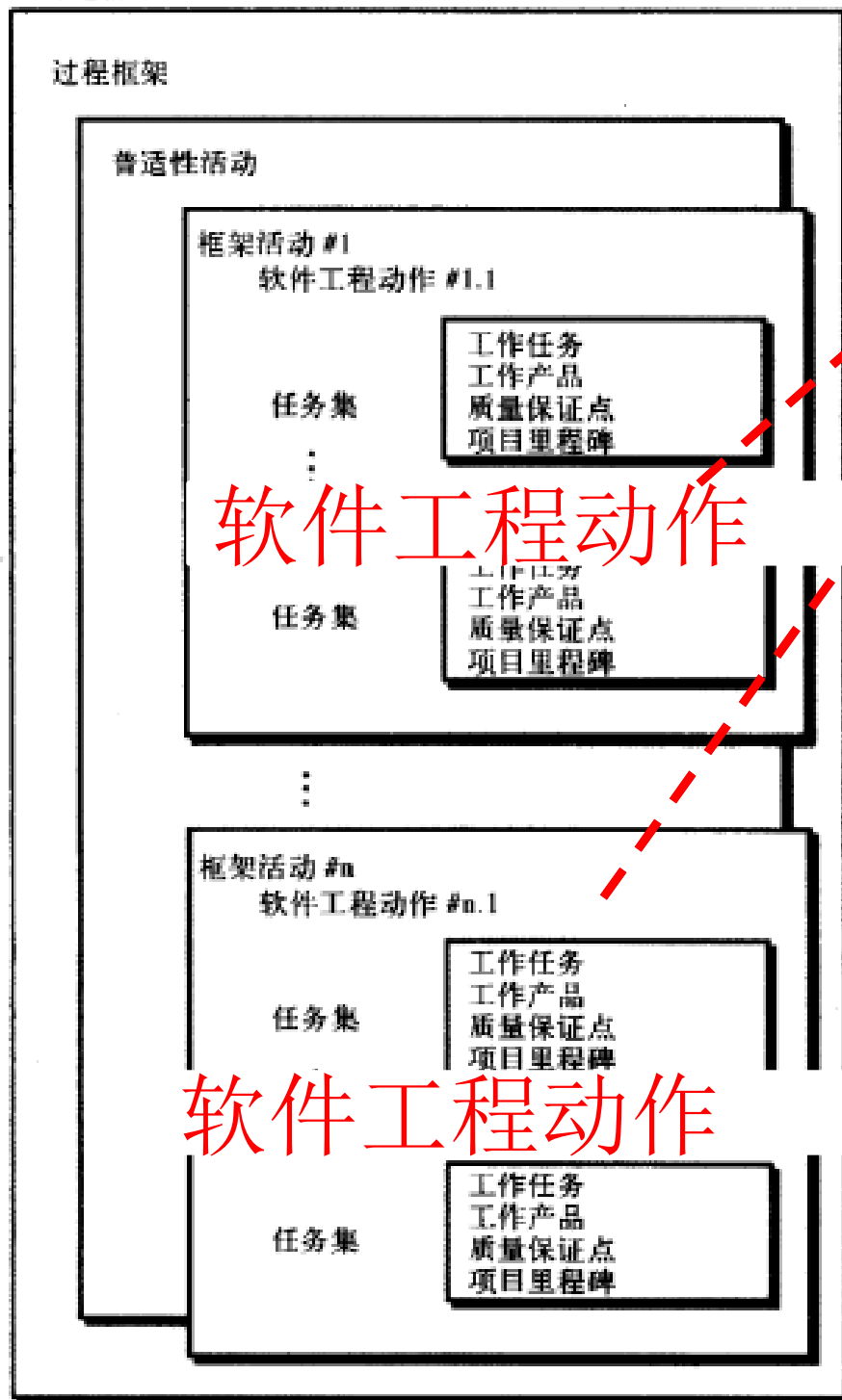
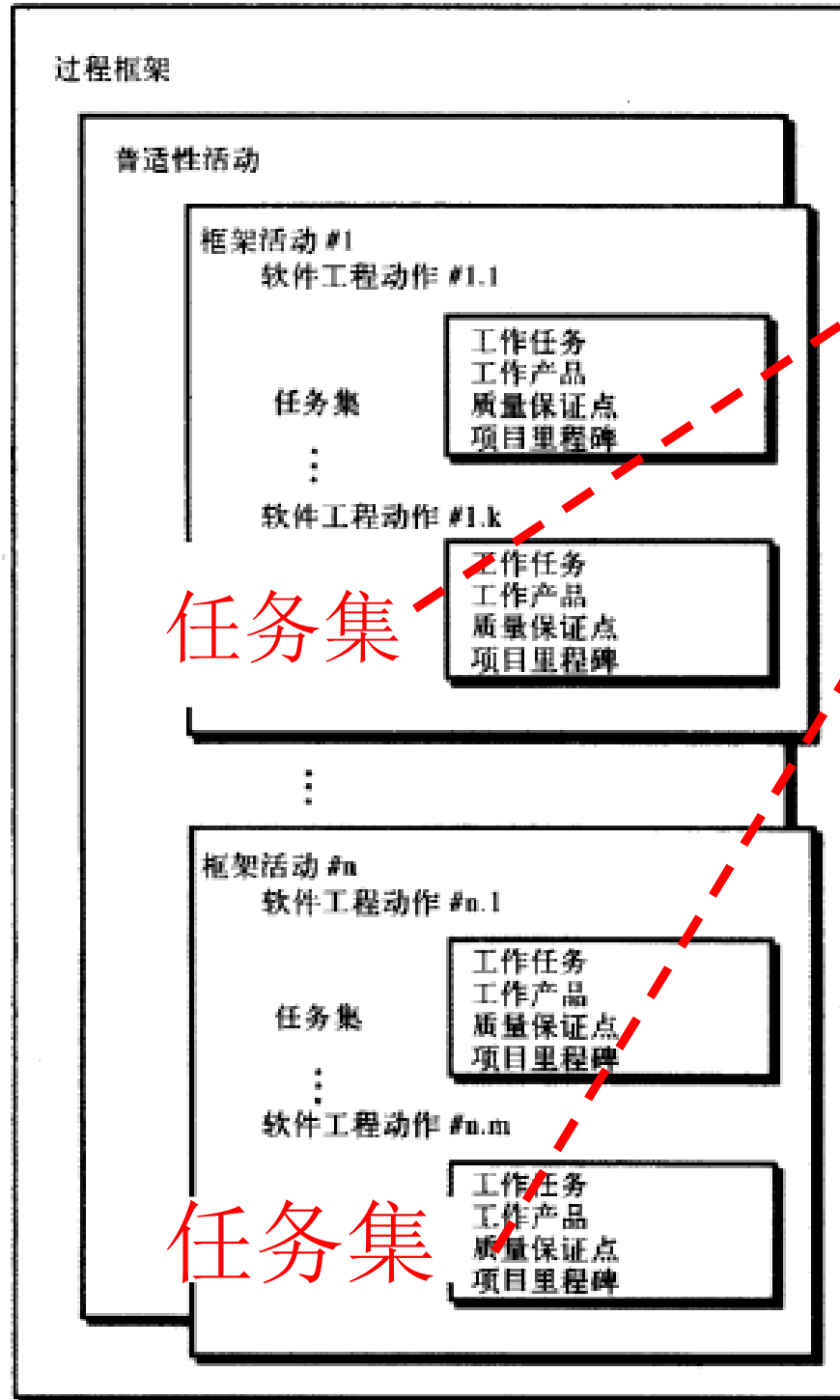


图  
2-1  
软件过程  
框架



每个动作包括一系列相互关联的**任务**，并产生一个关键的**工作产品**。比如“沟通”包含若干个软件 Engineering 动作，比如“获取需求”。

图  
2-1  
软件过程  
框架



定义了一个**达到一个软件工程的动作的目标**所需要完成的工作。

关注小而明确的目标，比如构建一个单元测试。

项目不同，任务集或不同

对于小型、相对简单的项目而言，获取需求的任务集可能包括：

1. 制定一个项目的共利益者列表
2. 邀请所有的共利益者成员参加一个非正式会议
3. 咨询每一个共利益者对于软件特征和功能的需求
4. 讨论需求。并确定最终的需求列表
5. 划定需求优先级
6. 把不确定的地方标记出来

对于大型、复杂的软件工程项目而言，可能有不同的任务集，例如：

1. 制定一个项目的共利益者列表
2. 和共利益者的每一个成员分别单独讨论，获取所有的需求。
3. 基于步骤2的调查，建立初步的功能和特征列表
4. 安排一系列促进需求获取的会议
5. 举行一系列会议
6. 在每次会议上建立非正式的用户场景
7. 根据共利益者的反馈，进一步细化用户场景
8. 建立一个修正的需求列表
9. 使用质量功能部署技术将需求划分优先级别
10. 将需求打包以便软件可以分批交付
11. 注意系统的约束和限制
12. 讨论系统的验证方法。

## 2.1 过程框架（续）

- 不同学者依据自己的理解将通用过程框架应用到侧重点不同的各个领域，提出了各自的用以描述过程的模型。

## 2.1 过程框架（续）

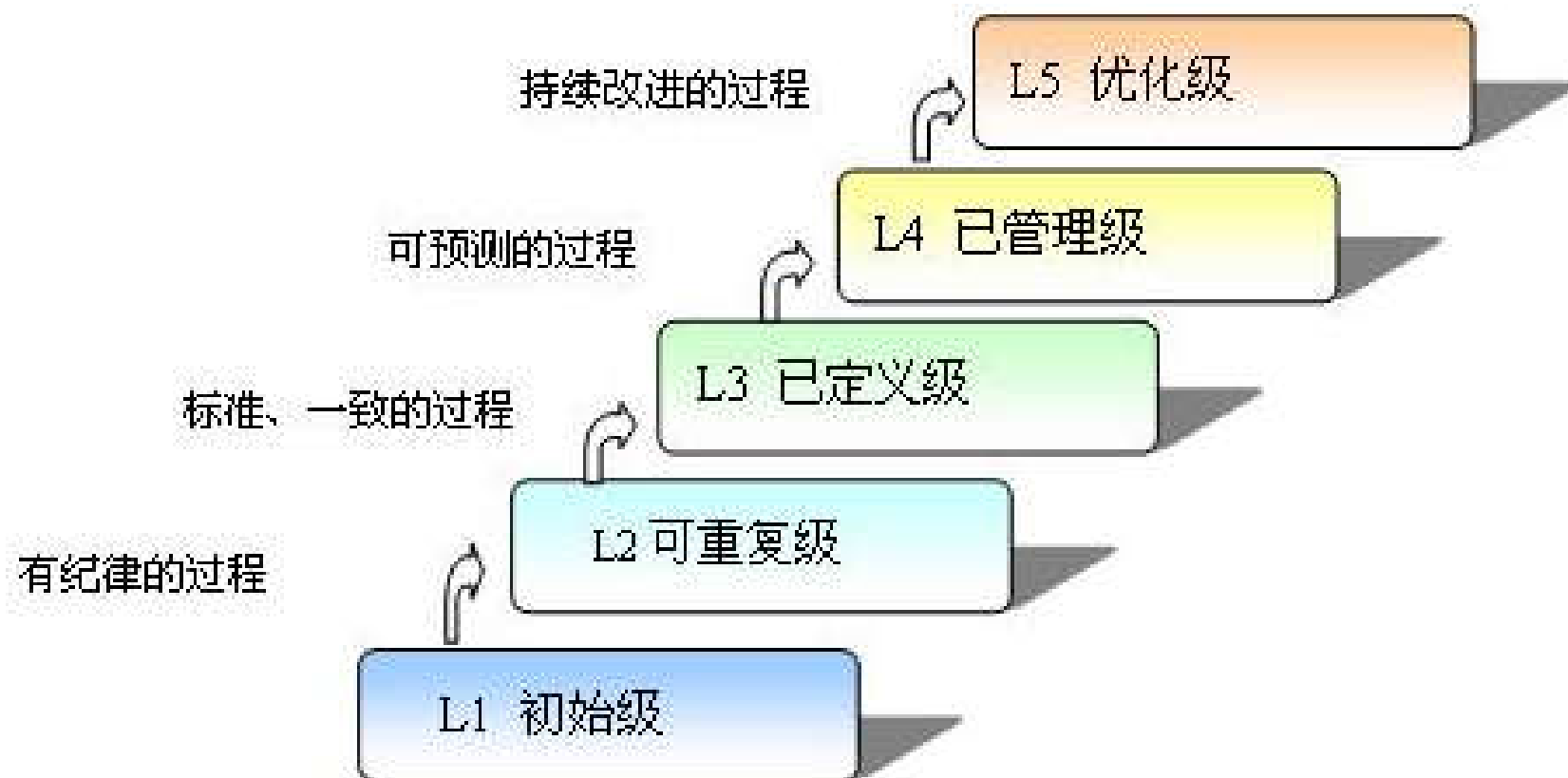
- 过程模型的**适用性**是成功的关键！不同的项目可能采用的项目过程有很大不同，主要体现在：
  - 活动、动作和任务的总体流程以及相互依赖关系；
  - 在每个框架活动中，动作和任务细化的程度；工作产品的定义和要求的程度；
  - 质量保证活动应用的方式；
  - 项目跟踪和控制活动应用的方式；
  - 过程描述的详细程度和严谨程度；
  - 软件团队所赋予的自主权；
  - 队伍组织和角色的明确程度。



## 2.2 能力成熟度模型集成

- 美国卡内基-梅隆大学软件工程研究所（SEI）在美国国防部资助下20世纪80年代末建立的能力成熟度模型集成（CMMI），用于预测软件开发组织所开发的系统和软件工程能力。
- 《2016中国CMMI咨询机构年度评选调查报告》

# CMM的5个能力成熟度等级



# CMMI

- CMMI定义了每一个过程域的“特定目标”，以及达到该目标所需的“特定实践”
  - **任务：**查相关资料，了解CMMI每级的目标和实践



# CMMI评估

兹证明

信息技术股份有限公司

**TRS Information Technology Co., Ltd.**

已经成功达到了

**CMMI-DEV (v1.3) 成熟度等级5级**

使用标准的CMMI评估方法 (v1.3)

评估编号: 3171



  
Masatoshi Hira, 评估师编号: 0400436-  
CMMI 研究院认证的高成熟度主任评估师

生效日期: 2019-4-30

失效日期: 2022-4-30



**CMMI Institute Partner**

级别	关注重点	项目管理	组织过程	工程管理	支持工程
2 Managed	项目基本管理	项目监督和控制 PMC		需求管理 REQM	过程 and 产品质量保证 PPQA
		项目计划 PP			配置管理 CM
		供应商合同管理 SAM			度量和分析 MA
3 Defined	组织级过程定义	集成项目管理 IPM	组织过程焦点 OPF	需求开发 RD	决策分析和解决方案 DAR
		风险管理 RSKM	组织过程定义 OPD	技术解决方案 TS	
			组织培训 OT	产品集成 PI	
				验证 VER	
				确认 VAL	
4 Quantitatively Managed	侧重量化和预防	量化项目管理 QPM	组织过程性能 OPP		
5 Optimizing	持续改进		组织改革和实施 OID		原因分析和解决方案 CAR

## 2.3 过程模式

- 所谓**过程模式**，比通用过程框架中定义的活动相对要具体些，包含这些通用的方面，但依据不同的领域，侧重点不一样，表述方法也不一样。（注意：模式的定义更宽泛，它可能适用于任何阶段，任何方面）

## 一个描述过程模式的模版：

- 模式名称
- 目的
- 类型（任务模式、步骤模式、阶段模式）
- 启动条件
- 问题
- 解决办法
- 结束条件
- 相关模式
- 已知应用实例

## 一个过程模式的例子

当利益共同体对工作成果有大致的想法，但对具体的软件需求不确认时，下述简化的过程模式描述了可采用的方法。

**模式名称。**原型开发。

**目的。**构造一个便于共利益者反复评估的模型（原型），以便识别和确定软件需求。

**类型。**阶段模式。

**启动条件。**在模式启动之前必须满足以下四个条件（1）确定共利益者；（2）已经建立起共利益者和软件开发队伍之间的沟通方式；（3）共利益者确定了需要解决的主要问题；（4）对项目范围、基本业务需求和项目约束条件有了初步了解。

**问题。**需求模糊或者不存在，但都清楚地认识到项目存在问题，且该问题必须通过软件解决。共利益者不确认他们想要什么，即他们无法详细描述软件需求。

**解决办法。**描述了原型开发过程，详见第3章。

**结束条件。**共利益者已经开发了一个软件原型，识别了基本的需求（例如，交互模式、计算特征、处理功能等）。随后，可能有两种结果：（1）原型系统可以通过一系列的增量开发成为软件产品；（2）原型系统被抛弃，采用其他过程模式建立了产品软件。

**相关模式。**以下模式与该模式相关：客户沟通，迭代设计，迭代开发，客户评估，需求抽取。

**已知应用实例。**当需求不确定时，推荐原型开发方法。



## 2.4 惯用过程模型

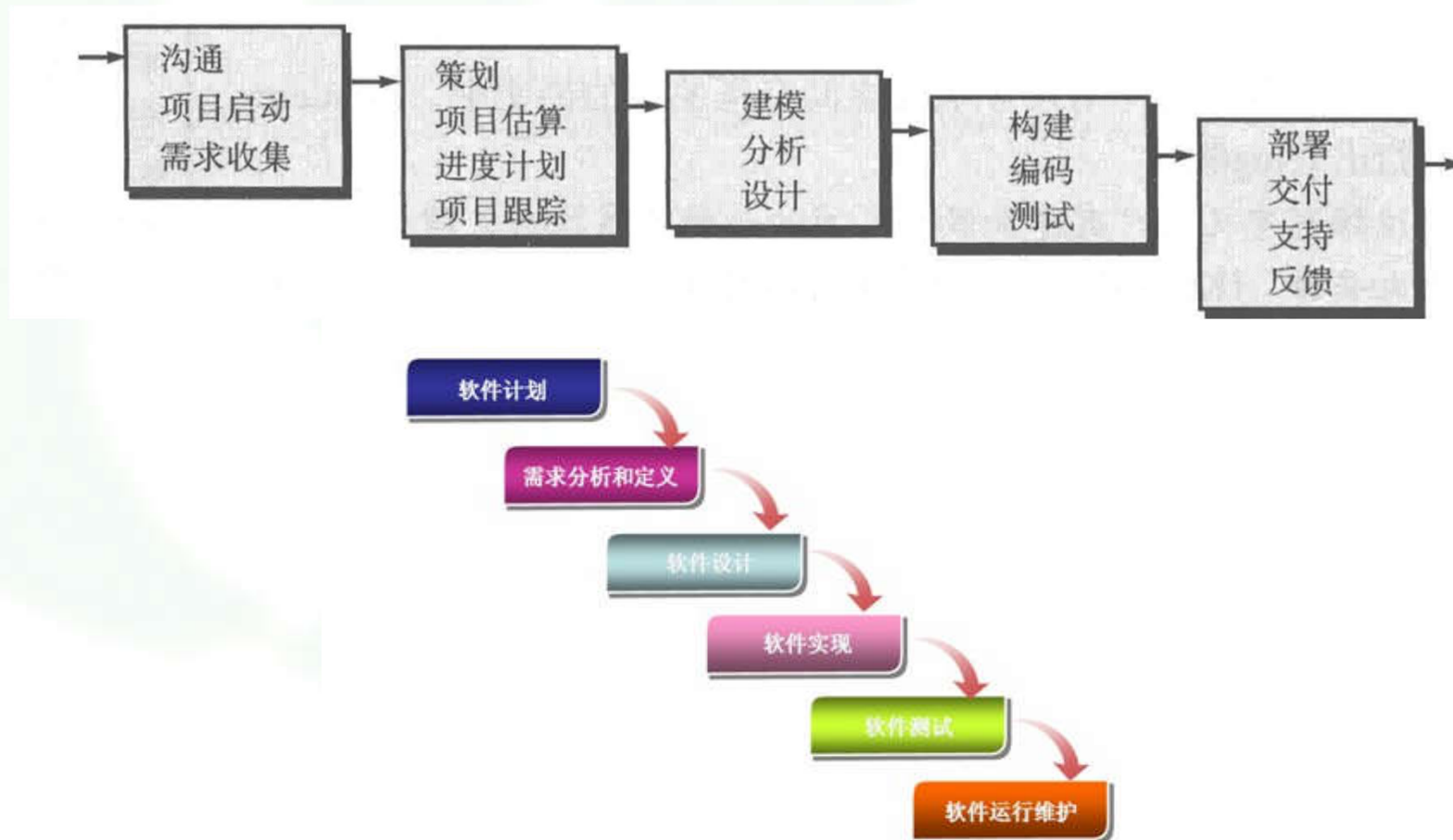
- 不同过程模型的共同特点：
  - 都支持通用过程结构，即都包含其五个要素：
    - 沟通、策划、建模、构建和部署
  - 规定了一整套的元素以及他们之间的相互关联的方式：
    - 框架活动、软件工程动作、任务、工作产品、质量保证以及每个项目的变更控制机制

## 2.4 惯用过程模型（续）

- 惯例过程模型包括：
  - 瀑布模型、增量过程模型（增量模型、RAD模型）、演化过程模型（原型开发、螺旋模型、协同开发模型）等。

## 2.4.1 瀑布模型

- 也称为线性模型或传统生存周期，V模型



# 瀑布模型（续）

- 适用范围？

- 通常发生在对一个已有系统进行明确定义的适应性调整和增强的时候
- 对于一个新的项目，需求必须是准确定义和相对稳定的

# 瀑布模型（续）

- 线性顺序模型特点：
  - 阶段间的顺序性和依赖性；
  - 文档驱动性；
  - 严格阶段评估；
  - 开发初期需要清楚全部需求；
  - 开发周期长、风险大。

瀑布模型的优点	<p>容易理解和计划</p> <p>适用于充分了解的小型项目</p> <p>分析和测试是顺序线性的</p>
瀑布模型的缺点	<p>不能很好地适应变化</p> <p>测试在过程的后期进行</p> <p>客户确认在最后阶段</p>

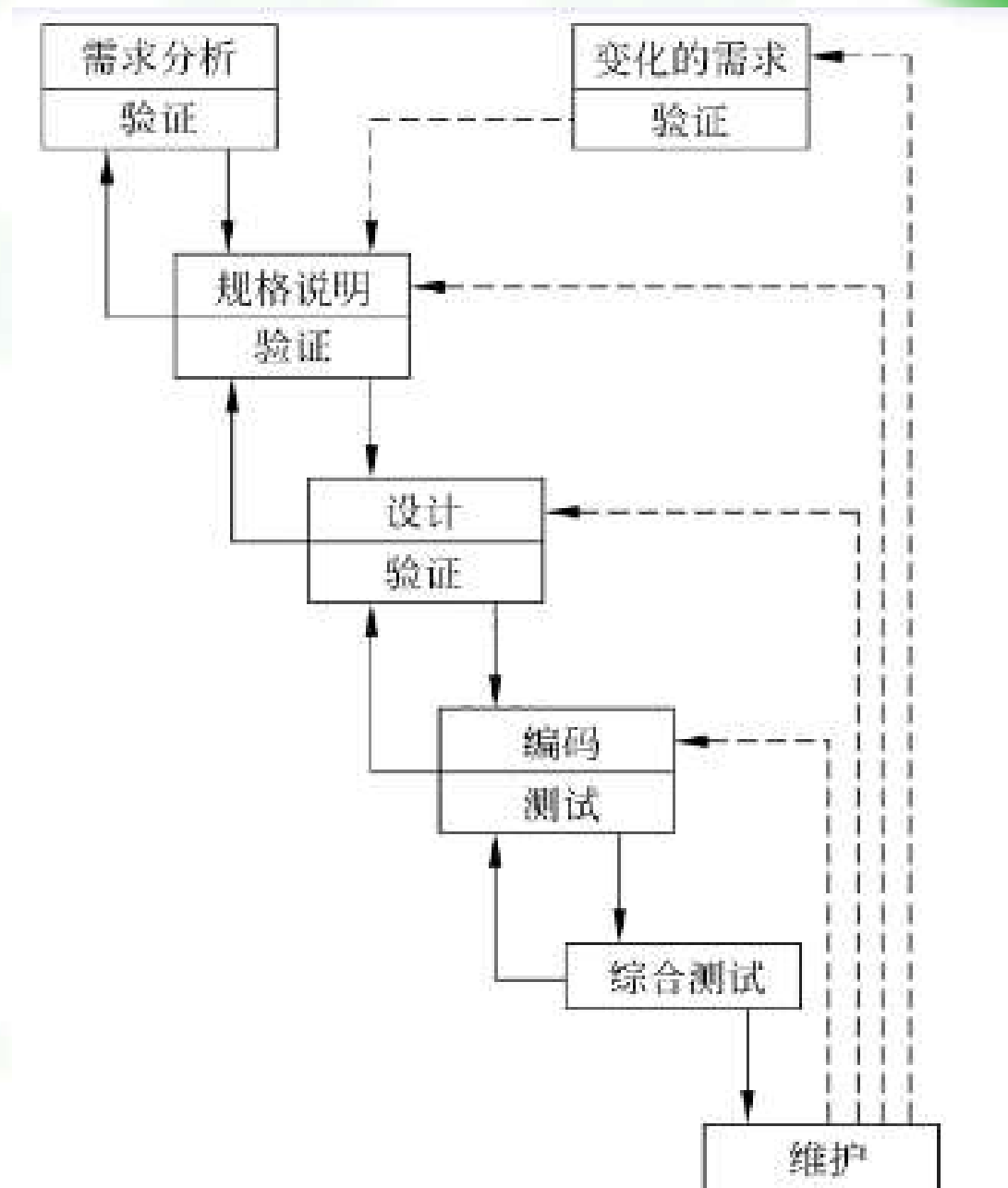


图 实际的带反馈环的瀑布模型

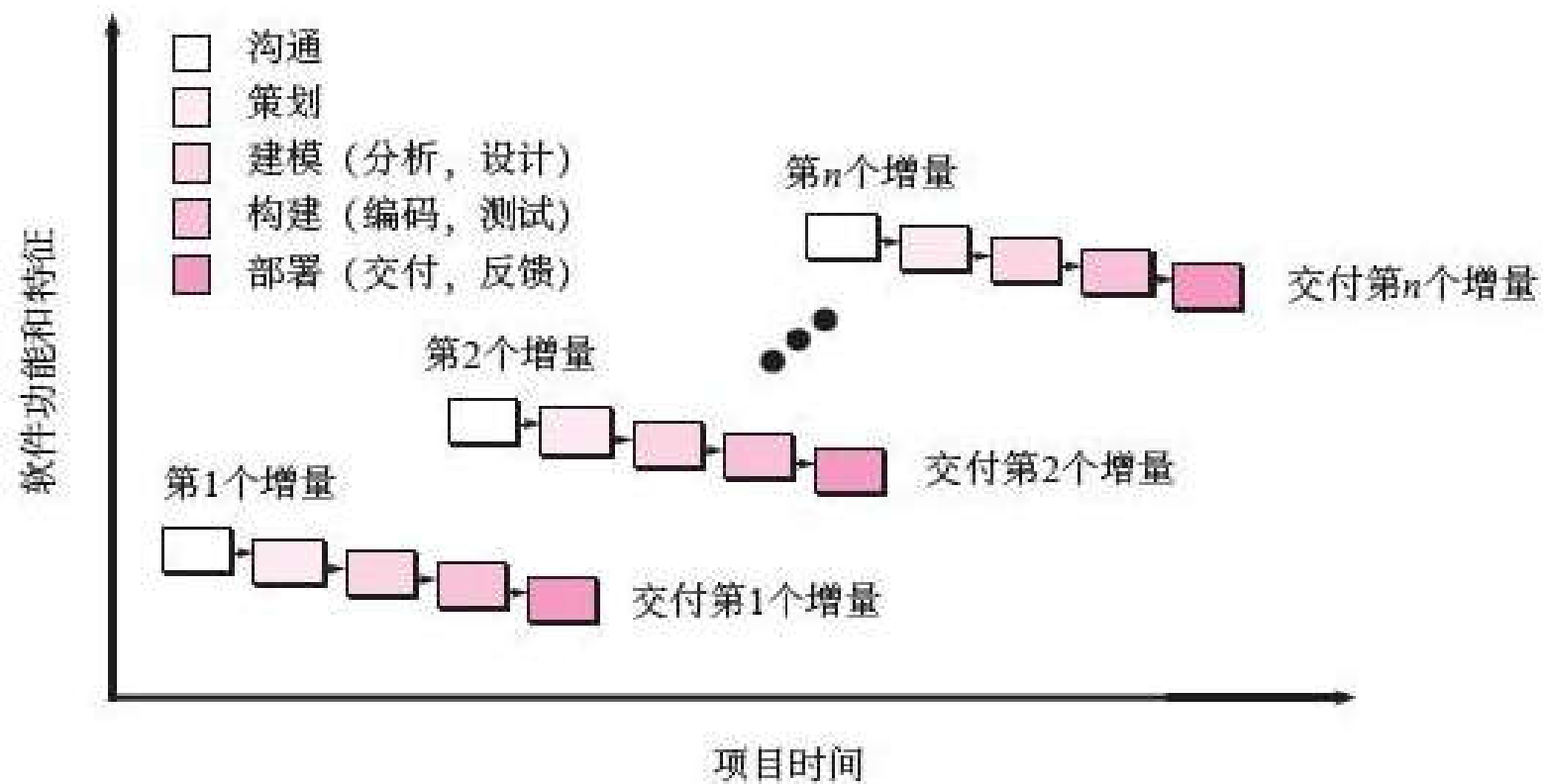
# 增量过程模型

- 增量过程模型包括：
  - 增量模型
  - RAD模型



# 增量模型

- 以迭代方式运用瀑布模型。



# 增量模型（续）

- 特点：

- 一般来讲，最重要的增量放在前面。
- 每次交付的增量产品都是可用的。
- 适合于功能可以划分，而且时间不紧迫的情况。
- 可以规避一定的风险。如有些技术还不稳定，将这部分放到后边。

# 增量模型（续）

- 例如：

- 采用增量模型开发的字处理软件：

- 在第1个增量中提供基本的文件管理、编辑和文档生成功能
    - 第2个增量中提供复杂的编辑和文档生成功能
    - 第3个增量中提供拼写和语法检查功能
    - 第4个增量中提供高级页面排版功能

增量模型优点	<ol style="list-style-type: none"><li>1. 将待开发的软件系统<b>模块化</b>，可以分批次的提交软件产品，使用户可以及时了解软件项目的进展。</li><li>2. 以组件为单位进行开发<b>降低了软件开发的</b>风险，一个开发周期内的错误不会影响到整个软件系统。</li><li>3. 开发顺序灵活，开发人员可以对组件的实现顺序进行优先级排序，先完成需要稳定的核心组件，当组件的优先级发生变化时，还能及时的对实现顺序进行调整。</li></ol>
增量模型缺点	<ol style="list-style-type: none"><li>1. 要求待开发的软件能够进行增量式的开发，否则会很麻烦。</li><li>2. 在软件开发过程中需求变化是不可避免的，增量模型的灵活性可以使其适应这种变化的能力大大优于瀑布模型和快速开发模型，但也很容易退化为边做边改模型，从而使软件过程的控制失去整体性。</li></ol>

# RAD模型

- 快速应用程序开发 (Rapid Application Development, RAD)是一种侧重于**短暂的开发周期**的增量软件模型。

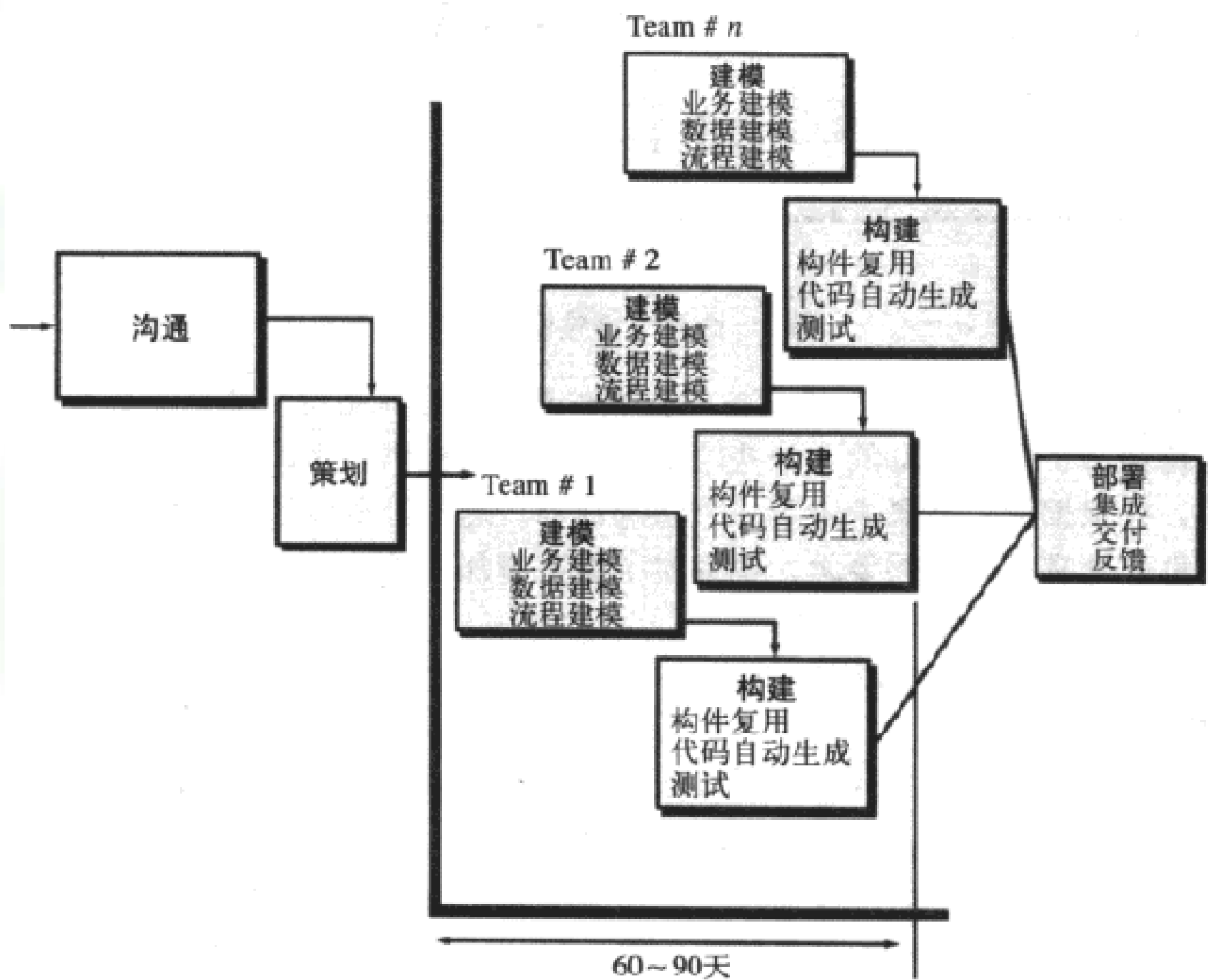


图3-3 RAD模型

# RAD模型（续）

- 瀑布模型的高速变体，通过基于构件的方法快速实现。
- 适于工期紧张，又可细分功能，还要有合适的构件。

# RAD模型（续）

- 缺点：

1. 需要投入更多的人力。
2. 各团队要紧密协作。
3. 只适应于特殊的系统，必须可以合理模块化。
4. 不适于高性能需求（若需调构件接口）
5. 系统需求灵活，现有构件不容易轻易满足。
6. 技术风险很高的情况下，不宜采用该模型。



# 演化过程模型

- 软件，类似于其他复杂的系统，会随着时间的推移而演化
- 软件有技术能力的限制，时间的限制，认识理解的限制，其它客观因素的限制。
- 演化模型也是一种迭代模型。

# 演化过程模型

- 演化模型包括：
  - 原型模型
  - 螺旋模型
  - 协同开发模型
  - .....

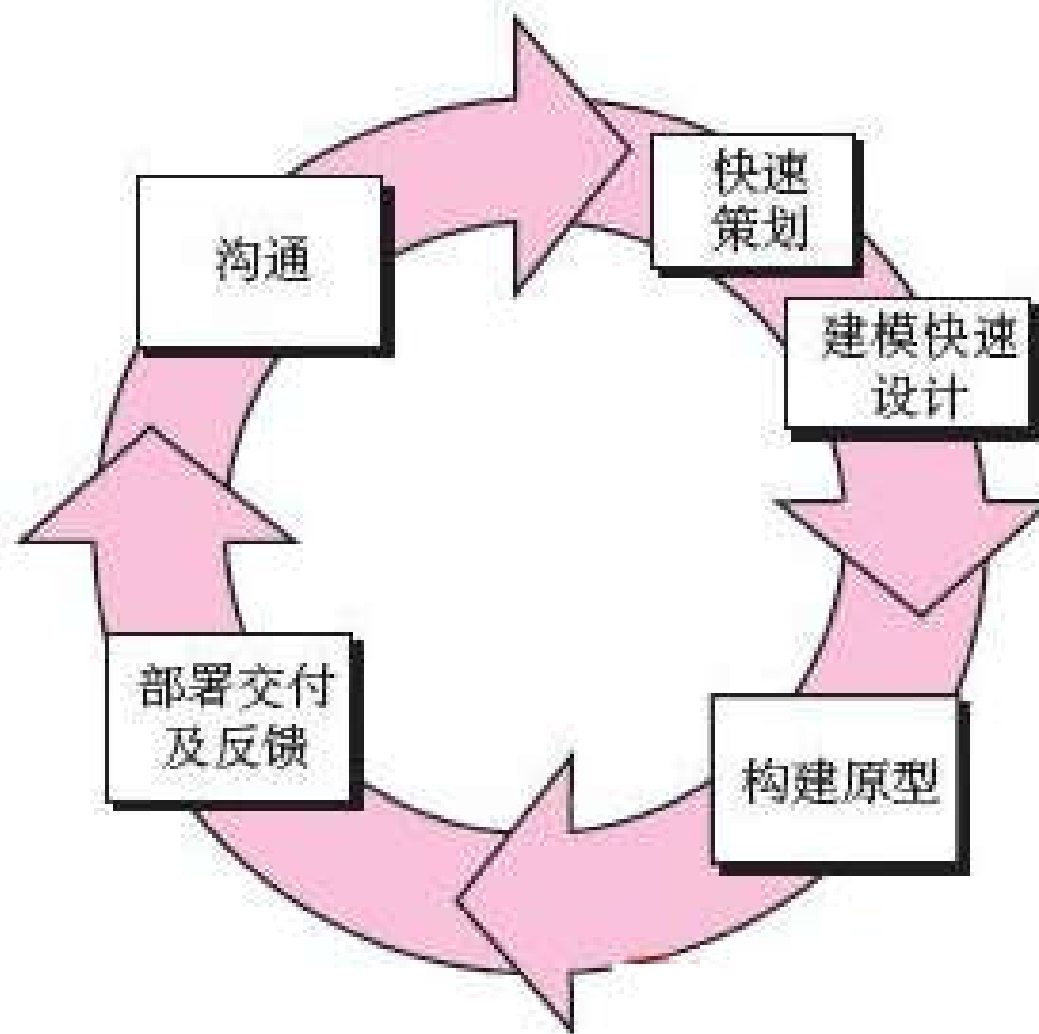
# 原型开发

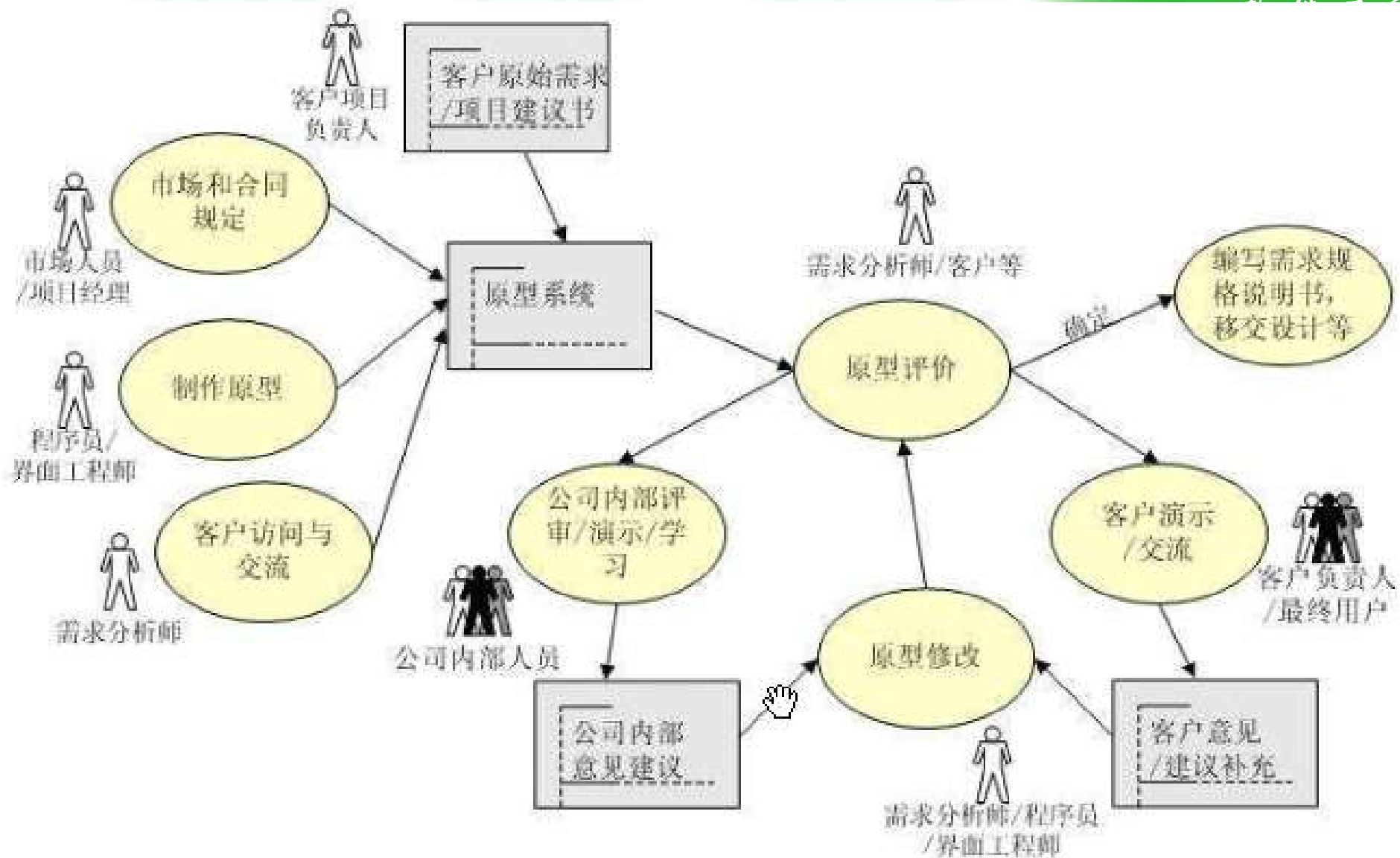
- 如果出现下面的情况，怎么办？
  - 客户也不能给出确切的要求
  - 开发人员对算法的效率、操作系统的兼容性和人机交互的形式不确定

## 原型

是一个循环的过程，所以也是迭代的过程。

# 原型开发（续）





图：原型法的处理过程

# 原型开发（续）

- 对原型的基本要求：
  - 体现主要的功能
  - 提供基本的界面风格
  - 展示比较模糊的部分，以便于确定或进一步明确，防患于未然
  - 原型最好是可以运行的，最少要在各主要功能模块之间能够建立相互连接

# 原型开发（续）

- 原型的处理方法：

- 抛弃型

- 在获取的明确需求的基础上，重新设计与开发
    - 成本相对高，小公司一般慎用

- 演化型

- 在原型的基础上继续开发

## 原型模型的优点

1. 变更需求对后续设计影响较小
2. 客户很早并频繁地参与其中
3. 对小型项目来说效果好
4. 产品失败的可能性降低

## 原型模型的缺点

1. 客户的参与可能造成进度延误
2. “提交”一个原型，可能造成初步完成的假象
3. 原型被抛弃导致工作白干了
4. 很难计划和管理



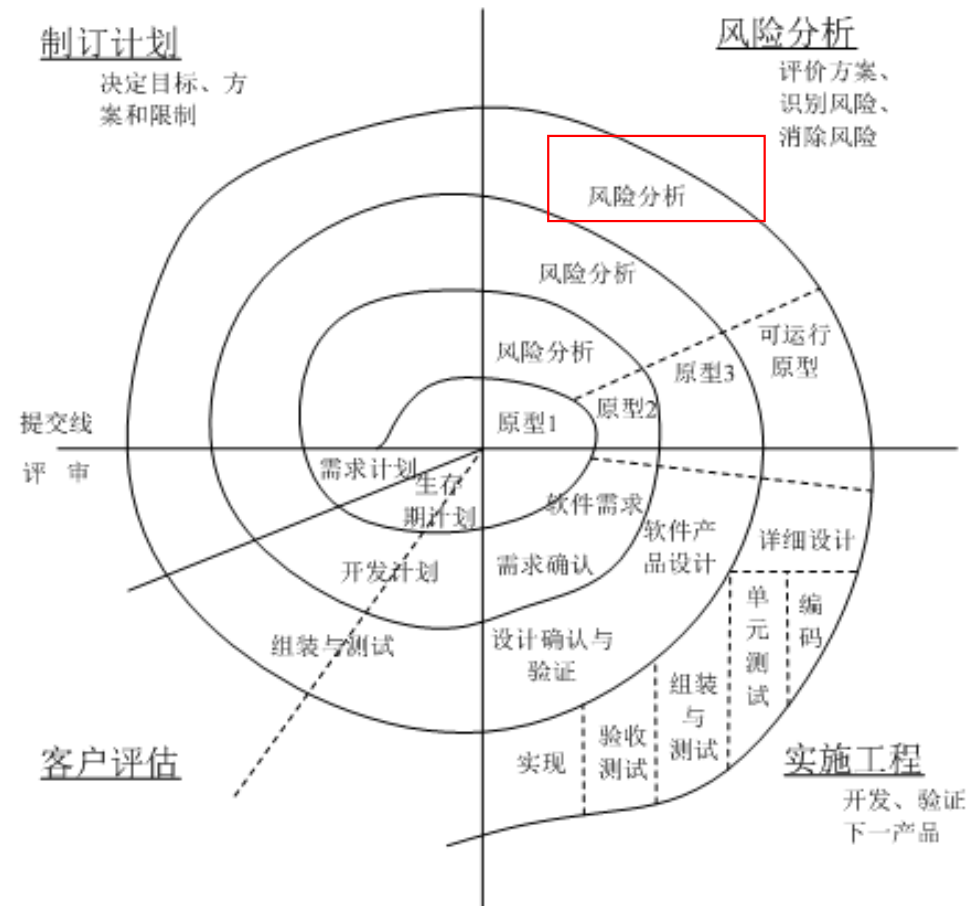
# 原型开发（续）

- 它与增量模型相比：
  - 增量模型在开发以前基本能确定系统的需求，虽然在以后的过程中也可能不断完善；原型开发适应于预先不太清楚系统的需求。
  - 增量模型的反馈可能较少，而原型开发需要不断的大量反馈信息。

# 螺旋模型

- 结合了原形的迭代性质和瀑布模型的系统性和可控性特点
- 风险驱动，引入非常严格的风险识别、风险分析和风险控制
- 早期迭代中可能是一个理论模型或原形

# 螺旋模型（续）



## 螺旋模型的优点

1. 有持续不断的客户参与
2. 开发风险得到控制
3. 适用于大型复杂项目
4. 适用于可扩展的产品

## 螺旋模型的缺点

1. 风险分析失败可能导致项目失败
2. 项目可能难于管理
3. 需要一个专家开发团队

# 螺旋模型（续）

- 螺旋模型与原型相比：
  1. 螺旋模型虽不像增量模型中对功能有明确界定，但有比原型要清晰一些。
  2. 螺旋模型的反馈要求持续于产品的整个生命期。
  3. 适合于大型软件的开发。

# 协同开发模型

- 又叫协同工程，并发开发模型。
- 定义了一个活动的网络，网络上每个活动、动作和任务同时存在。
- 过程网络中某一点产生的事件可以触发状态的转换。
- 可适用于所有类型的软件开发

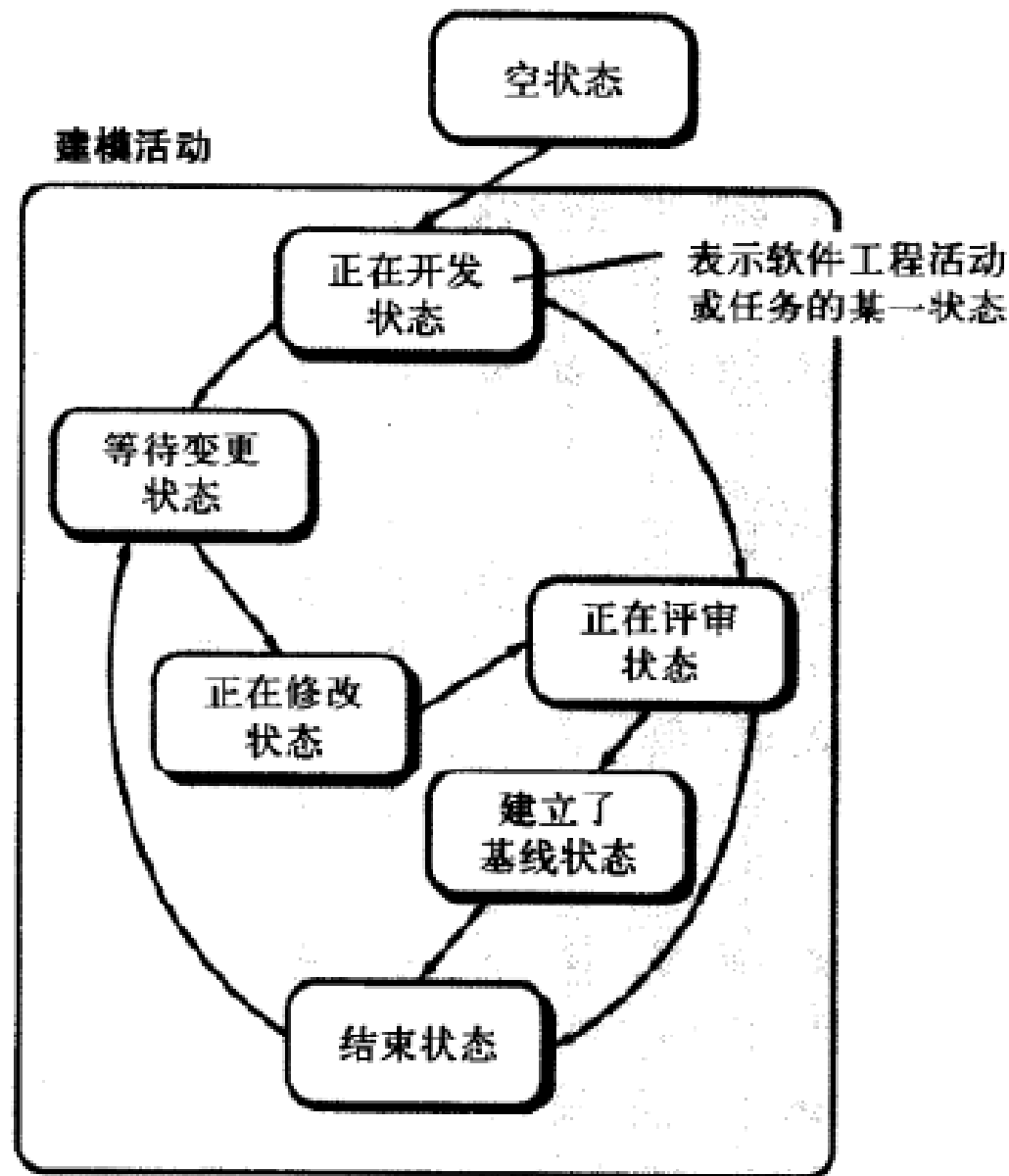


图3-6 协同过程模型的一个组成部分

# 演化过程评述

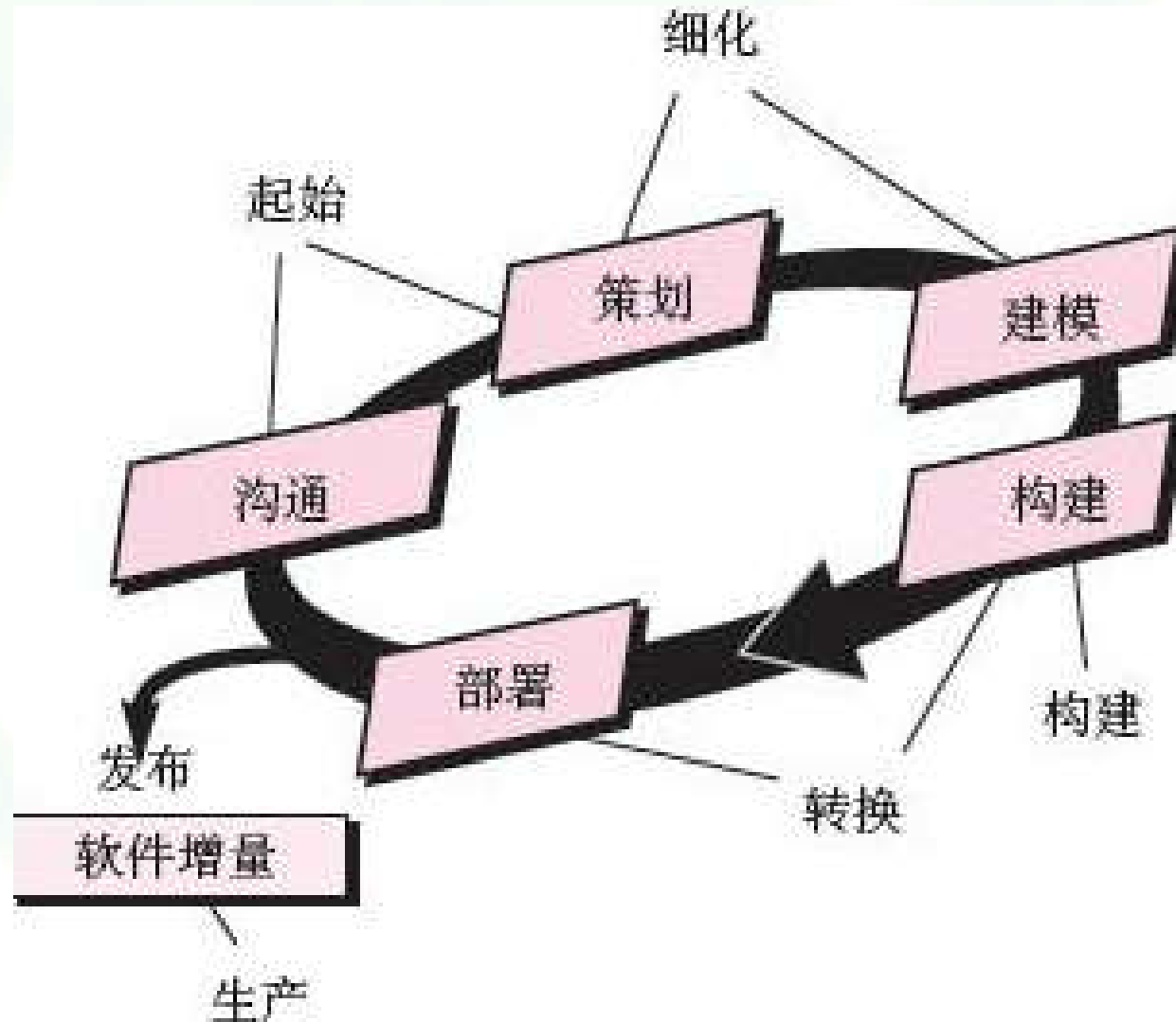
- 演化模型的初衷是采用迭代或者增量的方式开发高质量软件
- 用演化模型可以强调灵活性、可扩展性和开发速度
- 软件开发团队需要在严格的项目和产品参数与客户满意度之间找到一个平衡点

# 统一过程

- 试图将传统软件模型（惯例软件模型）和敏捷过程模型的优点结合起来，即统一起来。
- 一些术语：面向对象（Object-Oriented, OO), 面向对象分析（Object-Oriented Analysis, OOA), 面向对象设计（Object-Oriented Design, OOD).



- 统一过程包括：起始，细化，构建，转换，生产等步骤。



# 统一过程（续）

- 起始：
  - 包括客户沟通和策划活动
  - 此时的构架只是主要子系统及其功能、特性的试探性概括。

# 统一过程（续）

- 细化：
  - 包括用户沟通和通过过程模型的建模活动
  - 扩展体系结构以包括软件的五种视图：用例模型、分析模型、设计模型、实现模型和部署模型。

# 统一过程（续）

- 构建：
  - 与通过软件过程的构建活动相同。
  - 采用体系结构模型作为输入，开发或获取软件构件，使得最终用户能够操作用例。

# 统一过程（续）

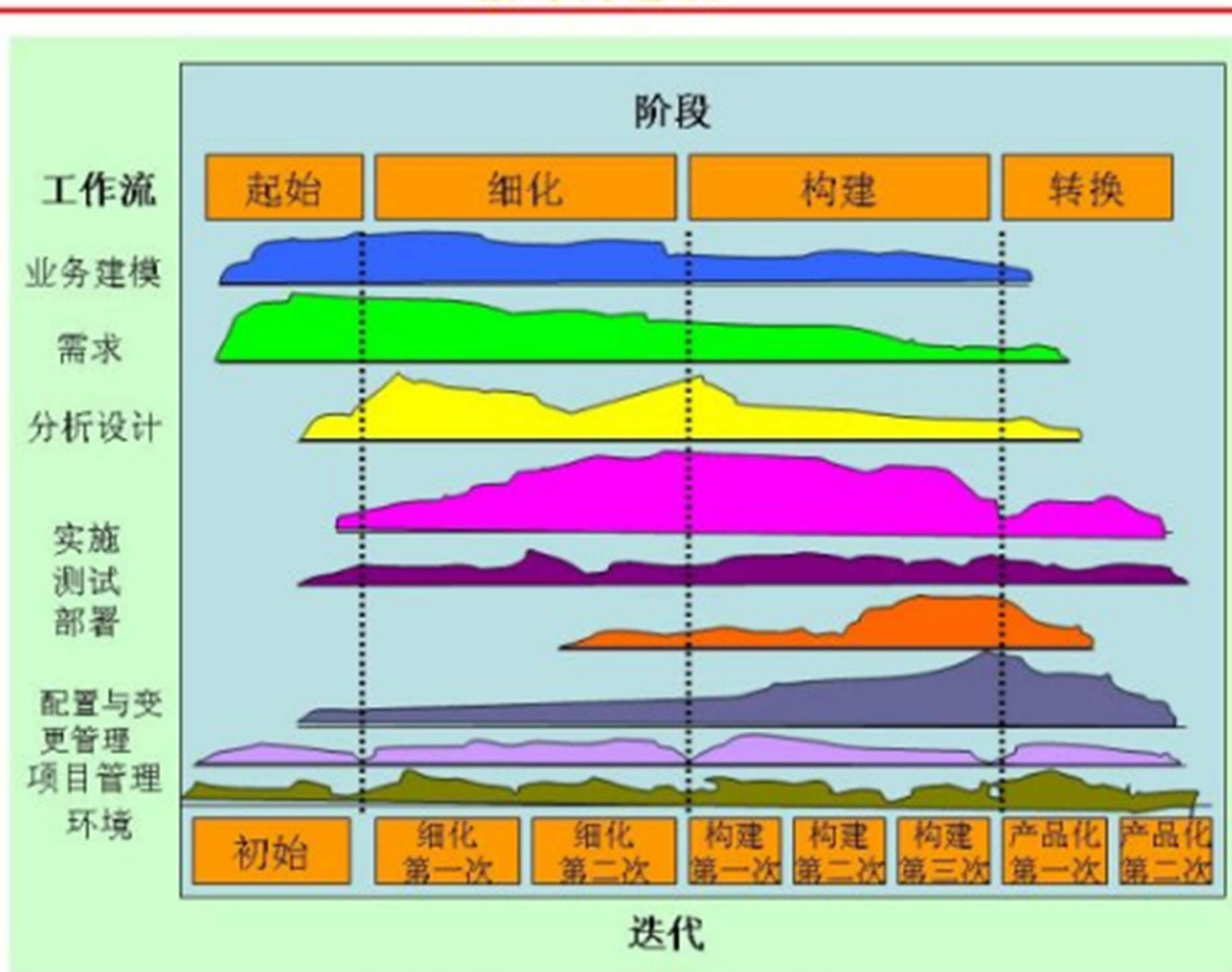
- 转换：
  - 软件被提交给最终用户进行**Beta**测试，用户反馈报告缺陷及必要的变更。
  - 另外，发布必须的支持信息：用户手册，用户指南及安装步骤等。
  - 结束时，软件增量成为可用的发布版本。

# 统一过程（续）

- 生产：
  - 与通过软件工程的部署一致
  - 提供运行环境支持，提交并评估缺陷报告和变更请求。

按时间组织

按内容组织



# 统一过程（续）

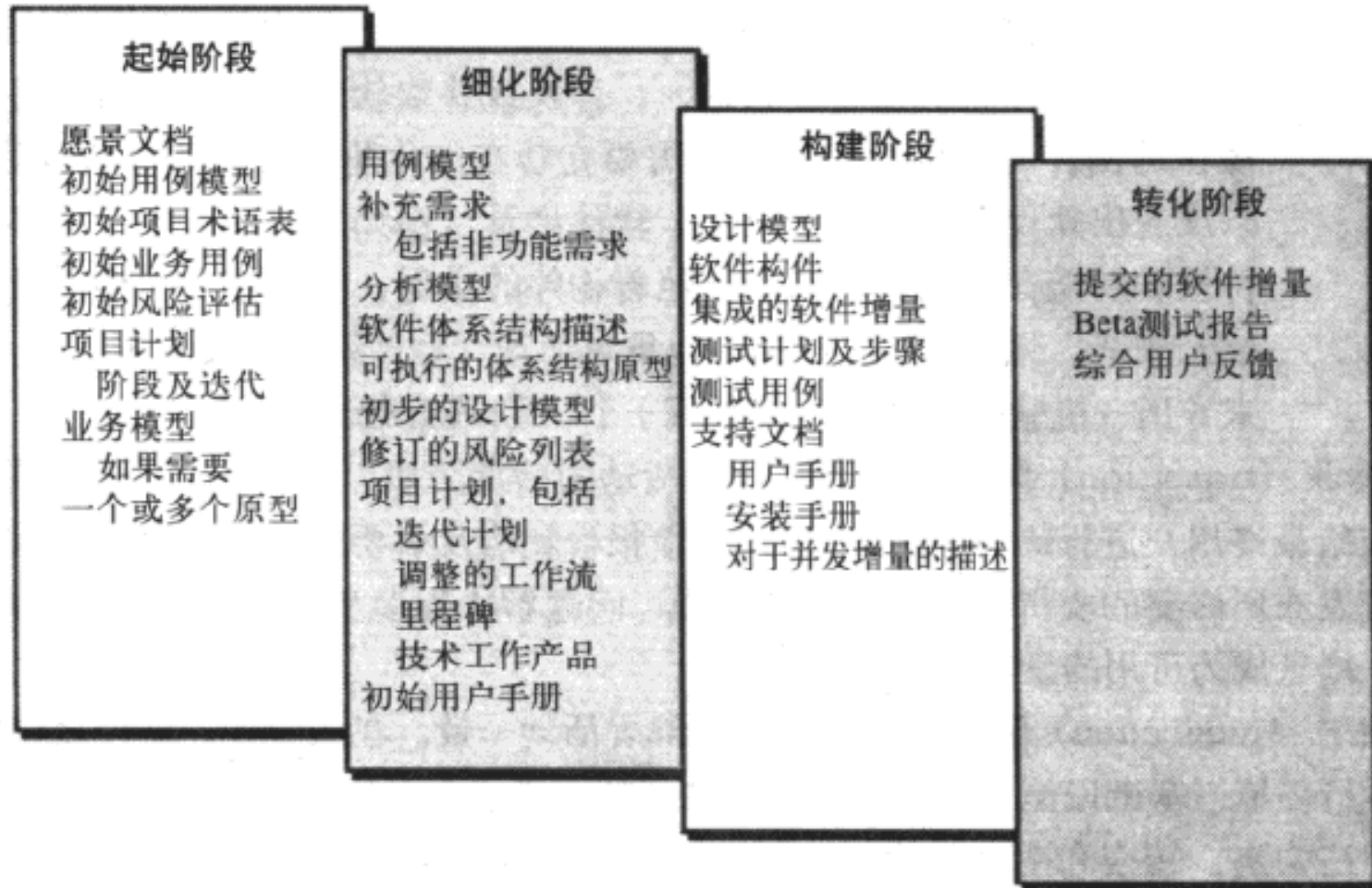


图3-8 在UP每一个阶段产生的主要工作产品



统一过程模型的优点	<ol style="list-style-type: none"><li>1. 重视质量文档</li><li>2. 有持续不断的客户参与</li><li>3. 适合需求变更的情况</li><li>4. 对维护项目非常有效</li></ol>
统一过程模型的缺点	<ol style="list-style-type: none"><li>1. 用例并不总是精确的</li><li>2. 具有复杂的软件增量集成</li><li>3. 阶段的重叠可能会带来问题</li><li>4. 需要一个专家开发团队</li></ol>

## 2.4.2 专用过程模型

- 具有通用过程模型的特点，只适应于一些特定的领域。
- 包括：
  1. 基于构件的开发
  2. 形式化方法模型
  3. 面向方面的软件开发等

# 基于构件的开发

- 利用预先打包的软件构件开发程序
- 什么是构件？
  - 没有统一的定义
  - Gartner Group定义：运行时软件构件是一个可动态绑定的、含一个或多个程序的**软件包**，它作为一个**独立单位**，通过运行时可辨别的文档化接口加以管理和存取
- 类似于螺旋模型，本质上是演化模型

# 基于构件的开发（续）

- 构件开发的步骤：
  - 对所需构件进行评估。
  - 考虑构件的集成。
  - 设计系统的软件框架。
  - 将构件放入框架。
  - 进行测试。

# 形式化方法模型

- 形式化方法模型的主要活动是生成计算机软件形式化的 $\text{数学规格说明}$ 。
- 特点：精密、准确。
- 缺点：难度大，成本高，可用人力资源少，用户不易理解，有时甚至无法完成。
- 方法：有穷状态机、Petri网、Z语言等。

# 面向方面的软件开发

- 将系统分成若干相对较独立的组成部分，这些部分称为方面。面向方面技术包括面向对象技术，比它大。
- 系统的方面包括用户接口、协调工作、发布、持续性、存储器管理、事务处理、安全、完整性等。
- 还不成熟。具有螺旋型和协同型的共同特点。

# 作业

- 理解：
  - 瀑布模型、原型模型、增量模型、构件模型、统一过程模型的适用范围、优点、缺点。
- 调研分析（交，分组讨论，每组交一份）：
  1. 描述一个适于采用瀑布模型的软件项目并分析适合采用瀑布模型原因
  2. 描述一个适于采用原型模型的软件项目并分析适合采用原型模型原因
  3. 描述一个适于采用增量模型的软件项目并分析适合采用增量模型原因
  4. 描述一个适于采用构件模型的软件项目并分析适合采用构件模型原因