



群聊：2025软件工程课程群



该二维码7天内(9月15日前)有效，重新进入将更新



北京大学

# 课程辅助学习平台——基于LLM的软件工程智能学习助手

## • 软件工程智能学习助手——辅助参考

- <http://162.105.16.170/sellm/> (校园网环境)
- 初始账号密码均为选课同学的学号
- 将大语言模型（LLM）智能体系统融入教学，提供个性化的学习辅助

The screenshot shows a web-based learning platform. On the left is a vertical sidebar with navigation links: '个人中心', '新建对话', '课程讲义' (highlighted in blue), 'UML演练场', '帮助', and '所有对话 (展开收起)'. At the top right, there is a page header with a back arrow, the number '5', and '/114', followed by a forward arrow. Below the header, the title '2.1 软件生存周期过程' is displayed in red. Underneath the title, a section titled '1、基本概念' is shown in red. It contains two bullet points: '• 软件生存周期' and '• 软件生存周期过程 (软件过程)'. Each point has a corresponding list of descriptions. At the bottom of the main content area, there is a logo of Peking University (北京大学) and a text input field asking '如何理解这一页中的软件生存周期?' (How to understand the software life cycle mentioned on this page?). A blue circular icon with a white arrow is positioned next to the input field.



北京大学

# 课程辅助学习平台——基于LLM的软件工程智能学习助手

## • 软件工程智能学习助手——辅助参考

- 为利用 UML 进行分析设计提供工具支持
- 包括用况图、类图、顺序图、活动图、状态图等
- 实时生成 UML 图，并与智能学习助手交互

个人中心

新建对话

课程讲义

UML 演练场

帮助

所有对话 (展开收起) ▾

选择模板: 用况图 ▾

```
@startuml
left to right direction
actor "Food Critic" as fc
rectangle Restaurant {
    (Eat Food) as UC1
    (Pay for Food) as UC2
    (Drink) as UC3
    UC1 <.. UC3 : <<extend>>
}
fc -- UC1
fc -- UC2
@enduml
```

如何完善这张用况图? |

发送



北京大学

# 复习第一次课、导入本次课的内容

## 软件工程

计算机类专业的一门核心专业课程，软件工程专业必须课  
设置本课程的目的

- 使学生掌握软件开发和维护的方法学，了解软件开发过程和软件项目管理基础知识
- 通过案例教学和课程实践培养学生软件开发和维护的能力
- 通过课程实践，培养学生软件项目管理的思想，即对一个软件项目的工作量、成本、进度和人员的计划和管理的思想
- 培养学生工程素质和团队精神，及综合实践的能力。
- 介绍前沿软件工程（人工智能、大数据、云计算、区块链、物联网驱动的软件工程），培养学生对软件工程的前瞻性和创新思维



软件工程课程是一门教授和指导学生们多人分组进行分布式协作、迭代式项目开发的课程，培养大家用系统化、工程化的思想、方法和技术解决复杂的问题。



北京大学

# 复习第一次课、导入本次课的内容

- 什么是软件?
  - 计算机系统中的程序及其文档。
    - 程序是对计算任务的处理对象和处理规则的描述；
    - 文档是为了理解程序所需的阐述性资料
- 软件的分类?
  - 系统软件、支撑软件、应用软件
- 软件有什么特点?
  - 软件是无形的、不可见的逻辑实体
  - 软件是设计开发的，而不是生产制造的
  - 软件是定制开发的
  - 软件是复杂的
  - 软件的开发成本高
  - 软件易于复制
  - 软件开发工作牵涉到很多社会因素
  - 软件质量要求较高
  - 软件的开发与运行都离不开相关的计算机系统环境



北京大学

# 复习第一次课、导入本次课的内容

- 什么是软件工程?
  - 60年代软件开发出现“软件危机”问题，为了解决“软件危机”提出软件工程概念！
  - 软件危机概括为以下特点：
    1. 软件质量差，可靠性难以保证；
    2. 成本难以控制，很少有在预算内完成的；
    3. 开发进度难以把握，周期拖得很长；
    4. 可维护性较差，维护人员和费用不断增加。

## 软件工程概念的提出

其目的是倡导以工程的原理、原则和方法进行软件开发，以解决软件危机。

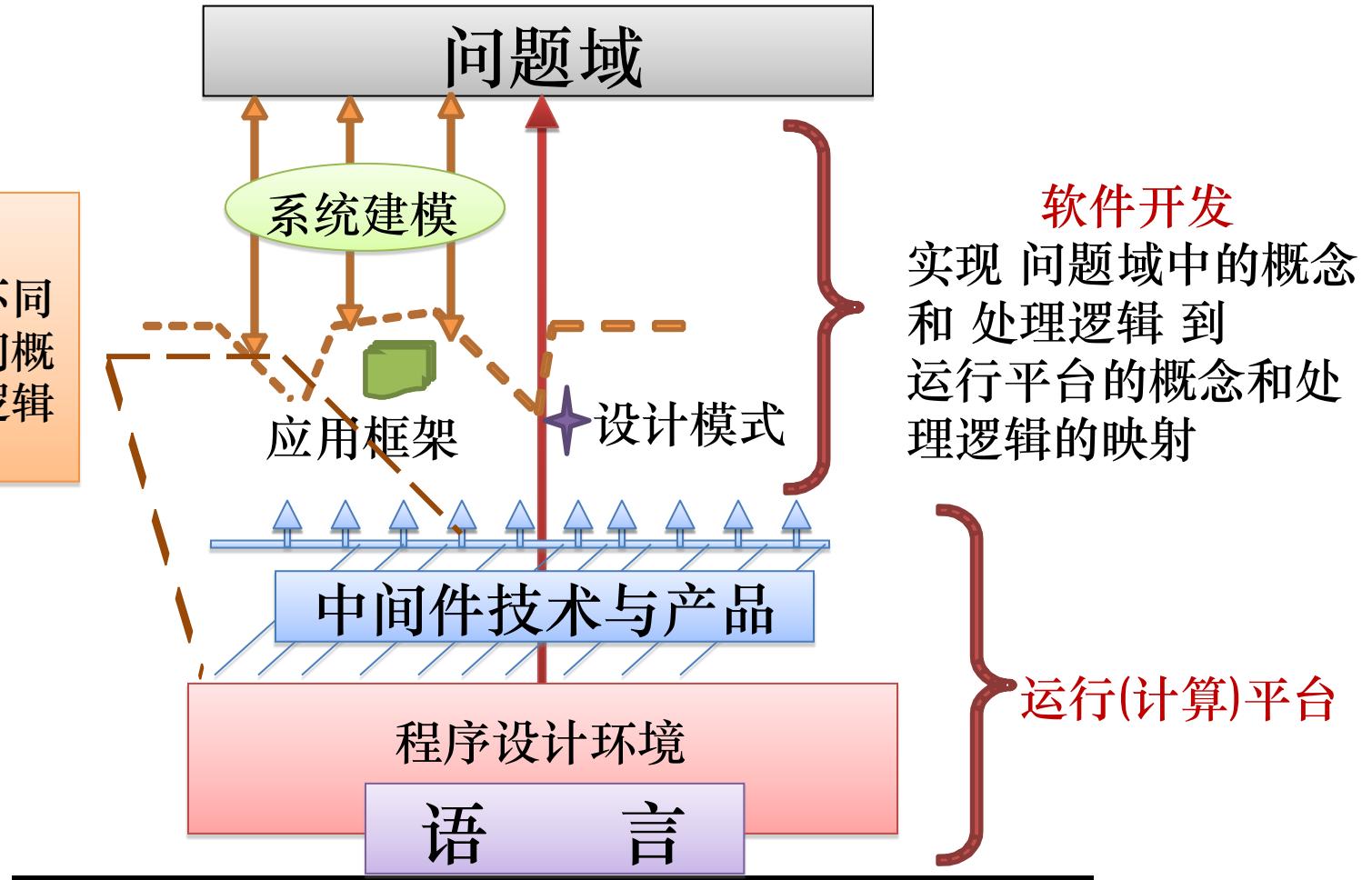


# 复习第一次课、导入本次课的内容

- 软件开发的本次是什么？

本质：

问题域到不同抽象层之间概念和计算逻辑的映射。

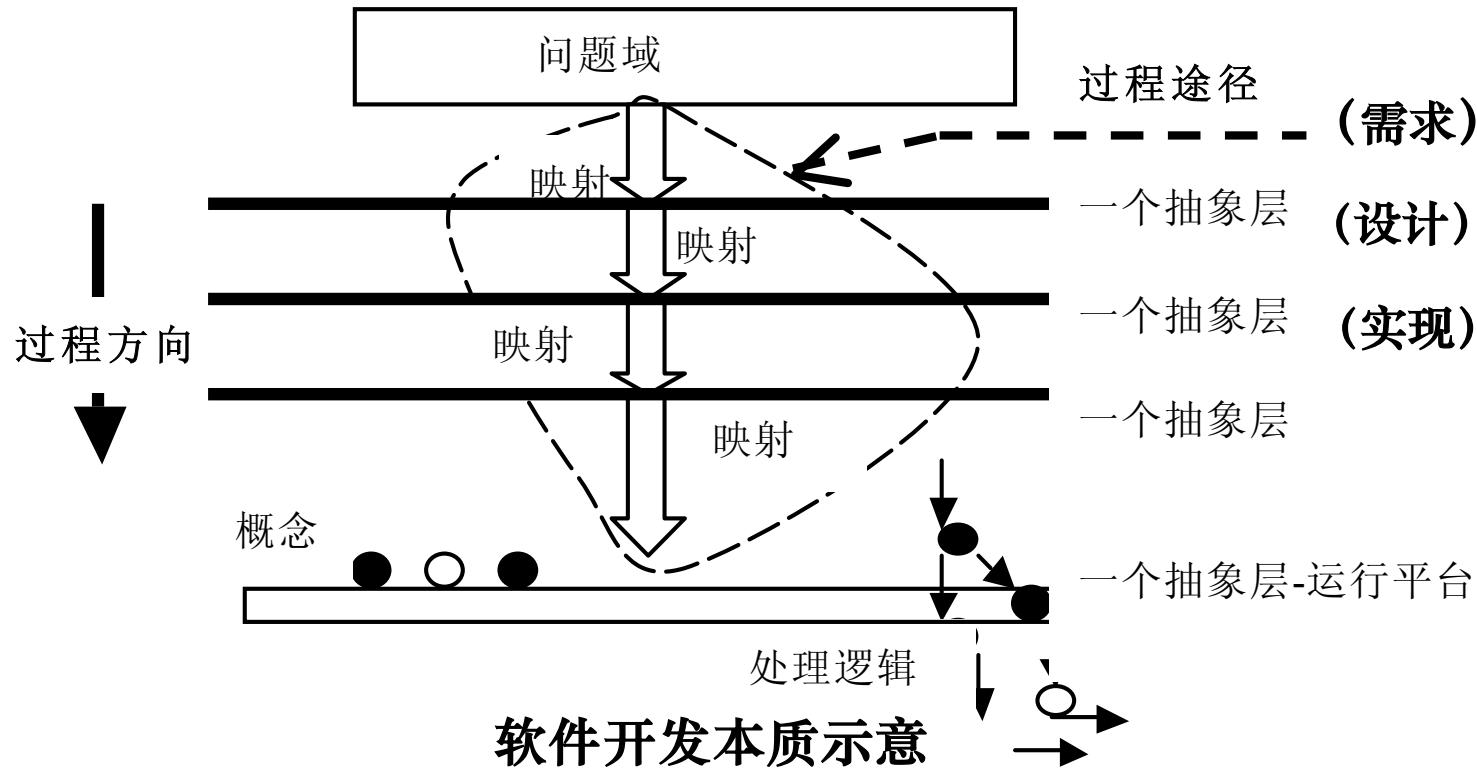


硬件系统，网络与操作系统 — 异构性



北京大学

# 复习第一次课、导入本次课的内容



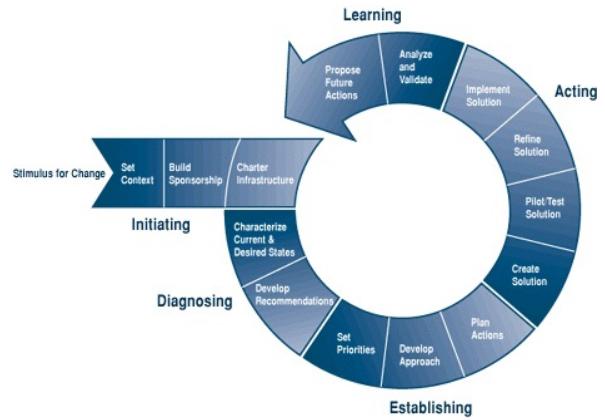
## 软件开发本质：

- 不同抽象层术语之间的“映射”
- 不同抽象层处理逻辑之间的“映射”



北京大学

# 第二章 软件过程



北京大学

# 软件过程

开发逻辑，是获取正确软件的关键。

- 软件开发有哪些活动—即要做哪些映射?
- 应如何正确组织开发活动，形成求解软件的逻辑?



北京大学

# 目录

- 2.1 软件生存周期过程
- 2.2 软件生存周期模型
- 2.3 软件过程改进

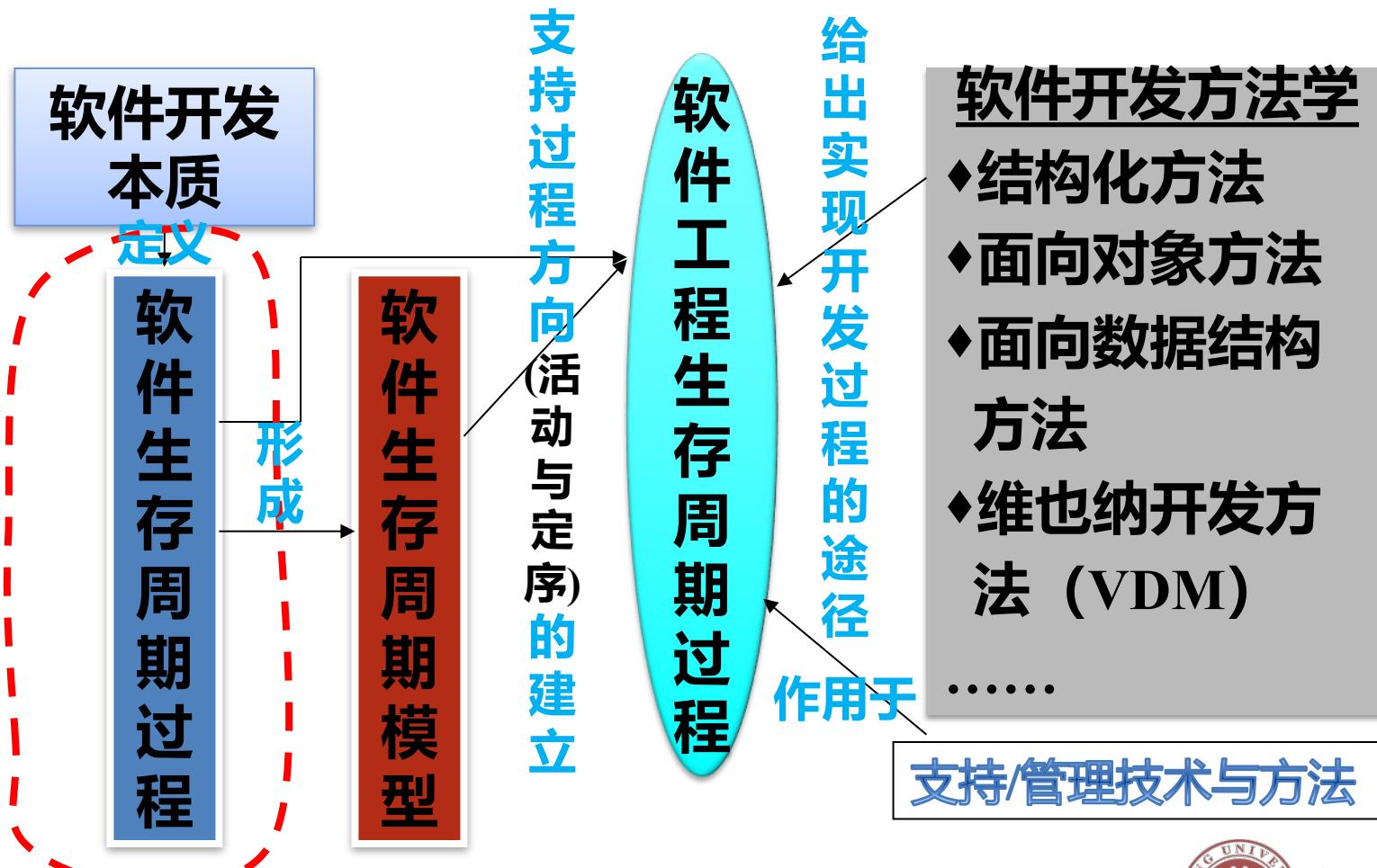


北京大学

## 2.1 软件生存周期过程

### 开发所涉及的活动

#### ---- 软件生存周期过程



北京大学

## 2.1 软件生存周期过程

### 1、基本概念

- 软件生存周期
  - 软件产品或系统的一系列相关活动的全周期。从形成概念开始，历经开发、交付使用、在使用中不断修订和演化，直到最后被淘汰，让位于新的软件产品。
- 软件生存周期过程（软件过程）
  - 软件生存周期中的一系列相关过程。
  - 为了表述软件开发需要做“什么活（映射）”，引入了以下三个概念：
    - **过程**是活动的集合，
    - **活动**是任务的集合，
    - **任务**是把输入转换成输出的操作。



北京大学

## 2.1 软件生存周期过程

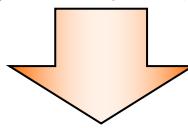
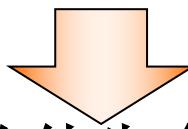
- **软件过程(process)**：活动的一个集合；
- **活动(activity)**：任务的一个集合；注：“软件过程”和“活动”相当于复合映射。
- **任务(task)**：将输入转换为输出的操作。注：“任务”相当于原子映射。



北京大学

## 2.1 软件生存周期过程

- ISO/IEC 12207-1995 软件生存周期过程
  - 系统地给出了软件开发所需的任务，即回答了“软件开发需要做哪些基本‘映射’”
  - ISO/IEC 12207-1995 补篇 1 (2002年)
  - ISO/IEC 12207-1995 补篇 2 (2004年)
- ISO/IEC 12207-2008 软件生命周期过程
  - 系统地给出了软件开发所需的任务，即回答了“软件开发需要做哪些基本‘映射’”
- ISO/IEC 12207-2017 软件生命周期过程
  - 系统地给出了软件开发所需的任务，即回答了“软件开发需要做哪些基本‘映射’”



北京大学

## 2.1 软件生存周期过程

### 2、过程分类

**按承担软件开发工作的主体，将软件生存周期过程分为四类：**

(1) 协议过程(agreement processes)

是指两个组织之间达成协议所需的活动集。

(2) 组织项目使能过程(organizational project-enabling processes)

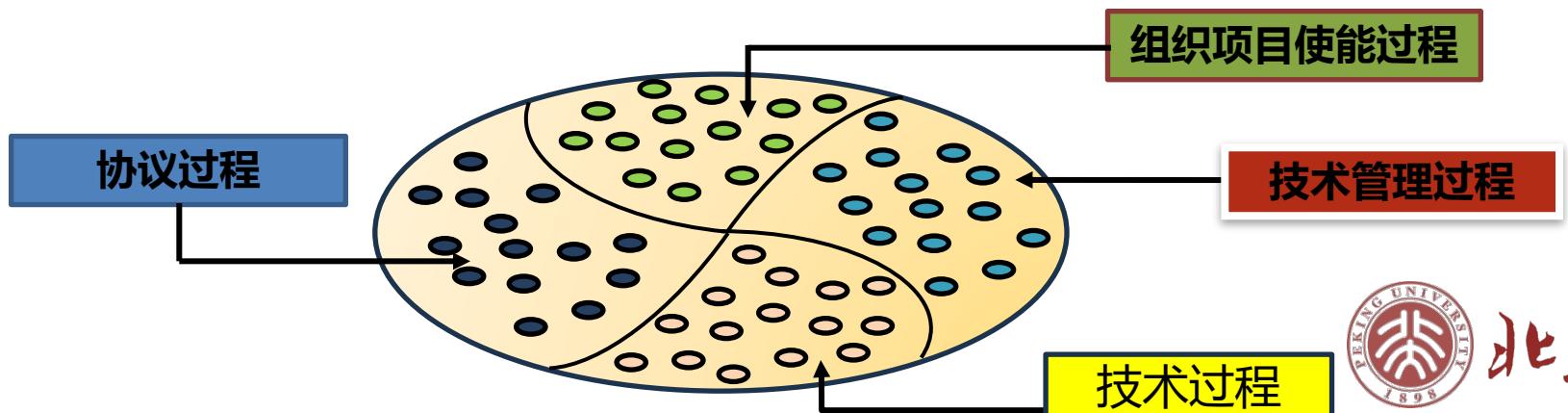
是指帮助确保组织获取和供应产品或服务的能力的一系列相关活动集。

(3) 技术管理过程(technical management processes)

是指用于管理计划，并控制从执行到实现的整个过程的活动集。

(4) 技术过程(technical processes)

是指从定义需求到处置退役产品的全过程中支持组织和项目职能的活动集。

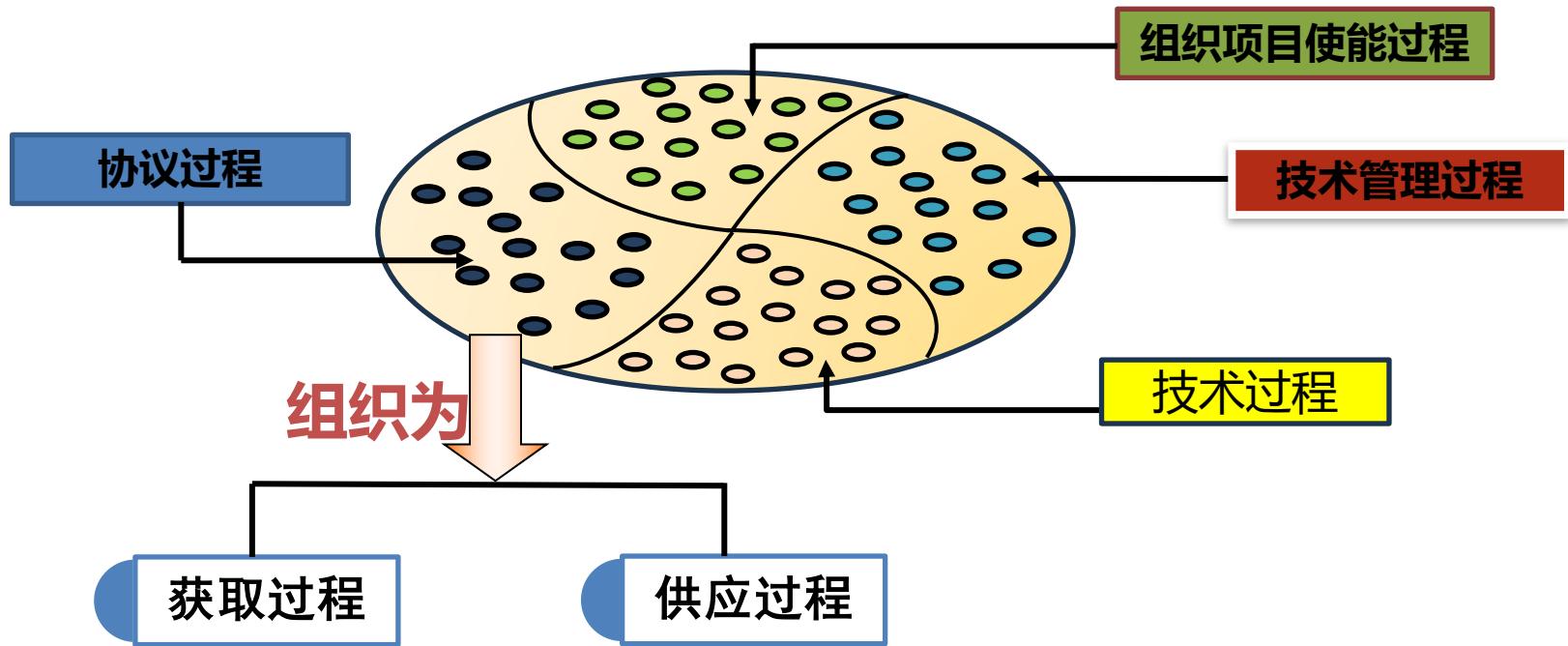


北京大学

## 2.1 软件生存周期过程

### (1) 协议过程

协议过程分为2个过程： 获取过程、 供应过程



北京大学

## 2.1 软件生存周期过程

- 1) 获取过程：**获取过程是获取者（**需方**）所从事的活动和任务，其目的是获得满足客户所表达的那些要求的产品和/或服务。该过程以定义客户要求开始，以接受客户所要求的产品和/或服务结束。
- 2) 供应过程：**供应过程是**供方**为了向客户提供满足需求的软件产品或服务所从事的一系列活动和任务，其目的是向客户提供一个满足已达成需求的产品或服务。



## 2.1 软件生存周期过程

### 例如1：供应过程

供应过程是供方为了向需方提供满足需求的软件产品或服务所从事的一系列活动和任务。

其目的是向需方提供符合约定要求的产品或服务。

该过程的启动，或通过为应答需方的招标书而开始编制投标书的决定，或通过与需方签订一项提供系统、软件产品或软件服务的合同。

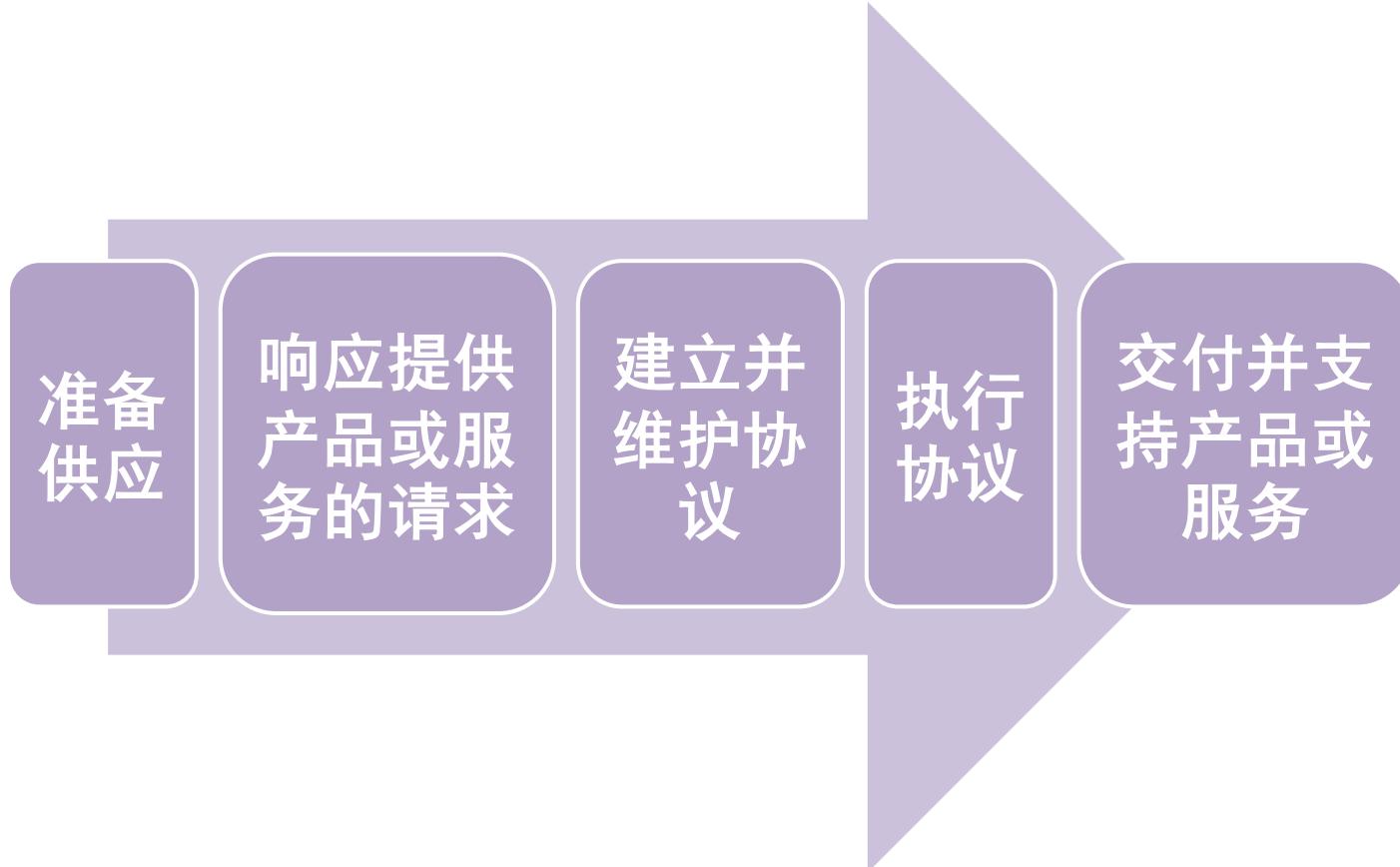
继之，确定为管理和保证项目所需的规程和资源，包括编制项目计划，执行计划，一直到将系统、软件产品或软件服务交付给需方为止。



北京大学

## 2.1 软件生存周期过程

该过程包括的基本活动为：



北京大学

## 2.1 软件生存周期过程

其中每一活动又包含一组特定的任务。例如“建立并维护协议”活动包括下述任务：

### (1) 与需方协商包含接受标准的协议。

注：该协议包括从书面合同到口头协议的不同形式。供应商确认要求、交付里程碑和验收条件是可以实现的，异常处理、协议变更管理程序和付款计划是可以接受的，并且它们为执行协议奠定了基础，没有不必要的风险。在谈判过程中讨论并解决问题，之后需方和供方接受协议条款并开始签订协议。



北京大学

## 2.1 软件生存周期过程

### (2) 确定协议的必要变更。

注：在请求更改协议时，需方或供方详细说明其规格、理由和背景。

### (3) 评估变更对协议的影响。

注：调查变更对项目计划、进度、成本、技术能力或质量的影响。

变更可以在现有协议内处理，也可能需要修改协议或新协议。

### (4) 必要时与需方协商协议。

注：协议条款的变更由供方和需方协商确定。这包括由于市场环境变化而发生的变更。针对初始协议以及任何变更的需要进行谈判。



北京大学

## 2.1 软件生存周期过程

### (5) 必要时更新与需方的协议。

注：协议修改的结果将纳入项目计划并传达给受影响的各方。

注：关于该过程的其它活动和任务，请参见相关的标准。

总的来说，成功实现该过程的结果是：

- A) 确定了产品或服务的需方。
- B) 生成了对需方请求的响应。
- C) 需方和供方之间达成了协议。
- D) 提供了产品或服务。
- E) 满足了协议中规定的供方义务。
- F) 按照协议的指示，转交了对所获得产品或服务的责任。

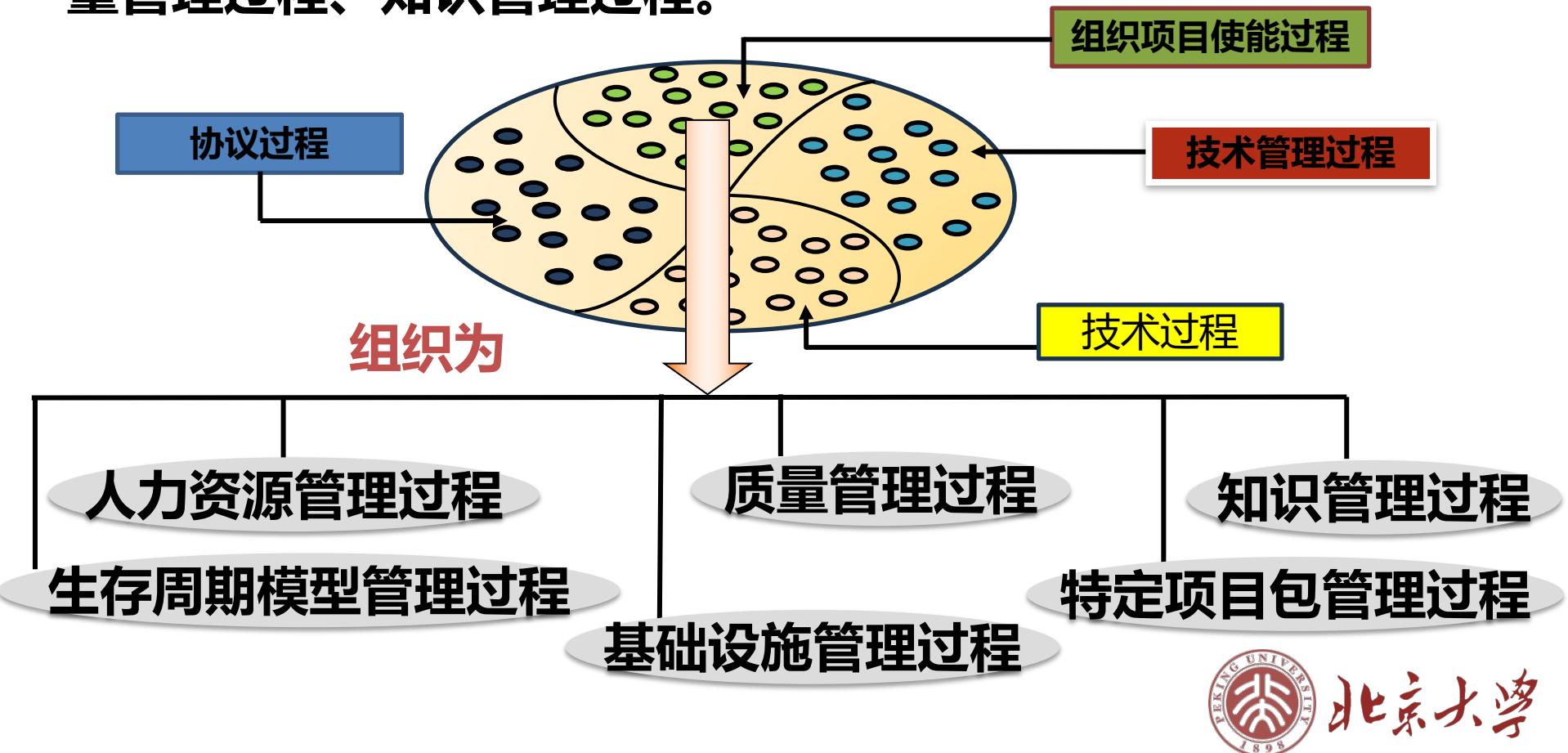


北京大学

## 2.1 软件生存周期过程

### (2) 组织项目使能过程

组织项目使能过程分为6个过程：生存周期模型管理过程、基础设施管理过程、特定项目包管理过程、人力资源管理过程、质量管理过程、知识管理过程。



北京大学

## 2.1 软件生存周期过程

- 1) 生存周期模型管理过程：**定义、维护并确保组织使用的政策、生存周期过程、生存周期模型和程序的可用性的活动。
- 2) 基础设施管理过程：**为项目提供基础设施和服务，以支持整个生存周期内的组织和项目目标的活动。
- 3) 特定项目包管理过程：**启动和维持必要、充分和合适的项目，以满足组织的战略目标的活动。
- 4) 人力资源管理过程：**为组织提供必要的人力资源，并根据业务需要保持其能力的活动。
- 5) 质量管理过程：**确保质量管理过程中的产品、服务等符合组织和项目质量目标，并实现客户满意度的活动。
- 6) 知识管理过程：**创造能力和资产，使组织能够利用机会重新应用现有知识的活动。



## 2.1 软件生存周期过程

### 例如2：生存周期模型管理过程

生存周期模型管理过程是定义、维护并确保组织所使用的政策、生存周期过程、生存周期模型和程序的可用性的过程。

包括3个活动：**建立过程**      **评估过程**      **改进过程**

其中的活动：**评估过程**

**主要任务：**

- ①监控整个组织的过程执行。
- ②对项目使用的生存周期模型进行定期审查。
- ③从评估结果中识别改进机会。

其中的活动：**改进过程**

**主要任务：**计划和实施改进，并告知利益相关方



北京大学

## 2.1 软件生存周期过程

总的来说，成功实施生存周期模型管理过程的结果是：

- A) 建立了管理和部署生存周期模型的组织政策和程序流程。
- B) 定义了生存周期政策、流程、模型和程序中的责任、问责和权限。
- C) 评估了组织使用的生存周期模型和过程。
- D) 实施了优先流程、模型和程序改进。



北京大学

## 2.1 软件生存周期过程

### 例如3：质量管理过程

质量管理过程是确保质量管理过程中的产品、服务等符合组织和项目质量目标，并实现客户满意度的过程

包括3个活动：

**计划质量管理   评估质量管理   执行纠正与预防措施**

其中的活动：**计划质量管理**

**主要任务：**

- ①建立质量管理政策、目标和程序。
- ②明确质量管理实施的职责和权限。
- ③定义质量评估标准和方法。
- ④为质量管理提供资源和信息。



北京大学

## 2.1 软件生存周期过程

其中的活动：执行纠正与预防措施

**主要任务：**

- ①当质量管理目标未实现时，计划纠正措施。
- ②当存在无法实现质量管理目标的充分风险时，计划预防措施。
- ③监督纠正和预防措施的完成，并通知利益相关者。



北京大学

## 2.1 软件生存周期过程

总的来说，成功实施质量管理体系的结果是：

- A) 确定并实施了组织质量管理政策、目标和程序；
- B) 建立了质量评价标准和方法。
- C) 向项目提供了资源和信息，以支持项目质量保证活动的运行和监测。
- D) 收集并分析了质量保证评估结果。
- E) 根据项目和组织结果改进了质量管理体系和程序。



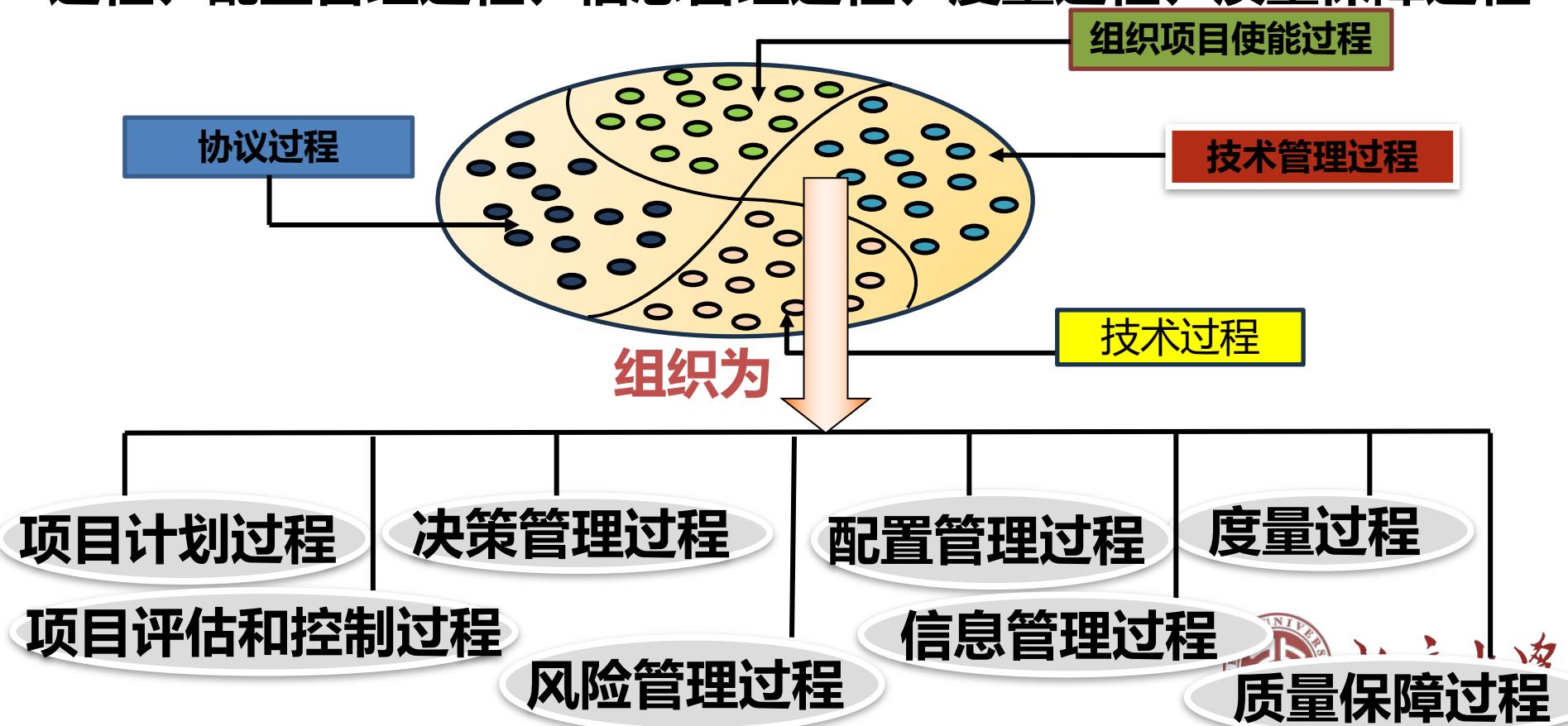
北京大学

## 2.1 软件生存周期过程

### (3) 技术管理过程

分为8个过程：

项目计划过程、项目评估和控制过程、决策管理过程、风险管理过程、配置管理过程、信息管理过程、度量过程、质量保障过程



## 2.1 软件生存周期过程

- 1) 项目计划过程**: 制定和协调有效可行的计划的过程。
- 2) 项目评估和控制过程**: 评估计划是否一致和可行, 确定项目的状态、技术和过程性能, 指导计划的执行, 以帮助确保工作性能符合计划和预算, 满足技术目标的过程。
- 3) 决策管理过程**: 提供一个结构化的分析框架, 以便在生存周期的任何时候客观地确定、描述和评估决策的一组备选方案, 并选择最有益的行动方案的过程。
- 4) 风险管理过程**: 系统产品或服务的整个生存周期中系统地处理风险的持续过程。
- 5) 配置管理**: 在生存周期中管理和控制系统元素和配置的过程。
- 6) 信息管理过程**: 生成、获取、确认、转换、保留、检索、传播和处置信息, 并将其发送给指定的利益相关者的过程。
- 7) 度量过程**: 收集、分析和报告客观数据和信息, 以支持有效管理, 并证明产品、服务和过程的质量的过程。
- 8) 质量保障过程**: 帮助确保组织质量管理过程在项目中的有效应用的过程。



北京大学

## 2.1 软件生存周期过程

例如4：项目计划过程

项目计划过程是制定和协调有效可行的计划的过程。

包括3个活动：**定义项目**   **计划项目和技术管理**   **激活项目**

**其中的活动：****定义项目**

**主要任务：**

- ① 确定项目目标和限制。
- ② 根据协议确定项目范围。
- ③ 使用定义的组织生存周期模型定义和维护由阶段组成的生存周期模型。
- ④ 基于可交付产品或软件系统的不断发展的体系结构，建立工作分解结构（WBS）。
- ⑤ 定义并维护将应用于项目的流程。



北京大学

## 2.1 软件生存周期过程

### 其中的活动：计划项目和技术管理

#### **主要任务：**

- ①根据管理和技术目标以及工作估算，定义并维护项目进度计划。
- ②定义生存周期阶段决策节点的实现标准、交付日期和对外部输入或输出的主要依赖关系。
- ③定义成本并计划预算。
- ④定义角色、职责、责任和权限。
- ⑤定义所需的基础设施和服务。
- ⑥计划采购从项目外部供应的材料、使能系统和服务。
- ⑦为项目和技术的管理和执行，包括评审，制定并传达计划。



## 2.1 软件生存周期过程

总的来说，成功实施项目计划过程的结果是：

- A) 确定了目标和计划。
- B) 定义了角色、职责、责任和权限。
- C) 正式请求并承诺了实现目标所需的资源和服务。
- D) 启动了项目执行计划。



北京大学

## 2.1 软件生存周期过程

### 例如5：项目评估和控制过程

项目评估和控制过程是评估计划是否一致和可行，确定项目的状态、技术和过程性能，指导计划的执行，以帮助确保工作性能符合计划和预算，满足技术目标的过程。

包括3个活动：**计划项目的评估和控制**   **评估项目**   **控制项目**



北京大学

## 2.1 软件生存周期过程

### 其中的活动：评估项目

#### **主要任务：**

- ①评估项目目标和计划与项目背景的一致性。
- ②根据目标评估管理和技术计划，以确定充分性和可行性。
- ③根据适当的计划评估项目和技术状态，以确定实际和预计成本、进度和性能差异。
- ④评估角色、职责、责任和权限的充分性。
- ⑤评估资源的充足性和可用性。
- ⑥使用度量的成就和里程碑完成情况评估进度。
- ⑦进行所需的管理和技术审查、审计和检查。
- ⑧监控关键流程和新技术。
- ⑨分析测量结果并提出建议。
- ⑩记录并提供评估任务的状态和结果。
- ⑪监控项目内的流程执行情况。



北京大学

## 2.1 软件生存周期过程

### 其中的活动：控制项目

#### 主要任务：

- ①采取必要措施解决已确定的问题。
- ②启动必要的项目重新规划过程。
- ③当因需方或供方请求的影响发生成本、时间或质量的合同变更时，启动变更行动。
- ④如果有正当理由，则授权项目进行下一个里程碑或事件。



北京大学

## 2.1 软件生存周期过程

总的来说，成功实施项目评估和控制过程的结果是：

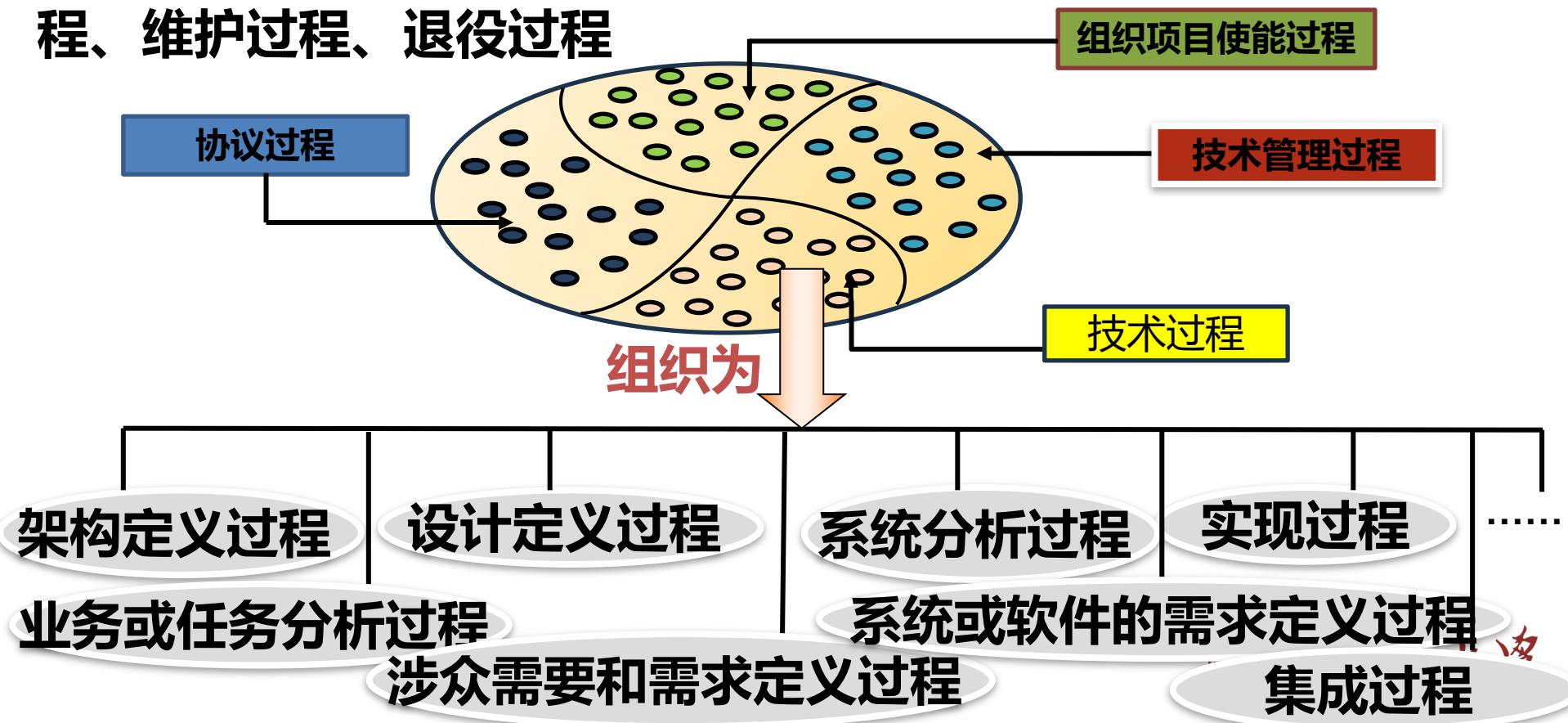
- A) 提供了绩效衡量或评估结果。
- B) 评估了角色、职责、责任和权限的充分性。
- C) 评估了资源的充足性。
- D) 进行了技术进度审查。
- E) 调查和分析了项目绩效与计划的偏差。
- F) 将项目状态告知了受影响的利益相关者。
- G) 当项目成果未达到目标时，确定并指导了纠正措施。
- H) 必要时启动项目重新规划。
- I) 授权了从一个预定里程碑或事件到下一个里程碑或事件的项目行动。
- J) 实现了项目目标。

## 2.1 软件生存周期过程

### (4) 技术过程

分为14个过程：

业务或任务分析过程、涉众需要和需求定义过程、系统或软件的需求定义过程、架构定义过程、设计定义过程、系统分析过程、实现过程、集成过程、验证过程、移交过程、确认过程、运行过程、维护过程、退役过程



## 2.1 软件生存周期过程

- 1) 业务或任务分析过程：** 定义业务或任务问题，刻画解决方案空间，并确定问题的潜在解决方案类别的过程。
- 2) 利益相关方需要和需求定义过程：** 定义利益相关方对系统的需求，表明系统应具备的能力的过程。
- 3) 系统或软件的需求定义过程：** 将所需的能力从面向利益相关方或用户的视图转化为具体解决方案的技术视图的过程。
- 4) 架构定义过程：** 生成系统架构备选方案，并以统一视图进行表达的过程。
- 5) 设计定义过程：** 提供关于系统及其元素的详细数据和信息，以达到与系统架构一致的实现的过程。
- 6) 系统分析过程：** 为技术理解提供严格的数据和信息基础，以辅助软件整个生存周期的各种决策的过程。



## 2.1 软件生存周期过程

- 7) 实现过程:** 实现系统元素的过程。
- 8) 集成过程:** 将系统元素合成为满足需求、架构和设计的软件产品的过程。
- 9) 验证过程:** 确保软件系统已实现规定的需求和功能的过程。
- 10) 移交过程:** 确保软件系统可以在运行环境中提供应满足的功能的过程。
- 11) 确认过程:** 提供客观证据，证明系统在使用时满足其业务或任务目标和利益相关方需求的过程。
- 12) 运行过程:** 软件交付后的使用过程。
- 13) 维护过程:** 维持系统正常提供服务的能力的过程。
- 14) 处置过程:** 结束一个或多个系统元素的存在的过程。



## 2.1 软件生存周期过程

例如6：系统或软件的需求定义过程

系统或软件的需求定义过程是将所需的能力从面向利益相关方或用户的视图转化为具体解决方案的技术视图的过程。

包括4个活动：

- 准备系统或软件的需求定义
- 定义系统或软件的需求
- 分析系统或软件需求
- 管理系统或软件需求



北京大学

## 2.1 软件生存周期过程

### 其中的活动：定义系统或软件的需求

#### 主要任务：

- ① 定义软件系统或元素需要执行的每个功能。
- ② 识别软件系统所需的运行状态或模式。
- ③ 定义必要的实施约束。
- ④ 识别与风险、软件系统的关键性或关键质量特征相关的需求。
- ⑤ 定义系统或软件需求和需求属性，包括数据元素、数据格式以及数据保存要求，用户界面和用户文档以及用户培训，与其他系统和服务的接口，功能和非功能特性（包括关键质量特性和成本目标）等。



北京大学

## 2.1 软件生存周期过程

其中的活动：分析系统或软件需求

**主要任务：**

- ① 分析整套系统或软件需求。
- ② 定义能够评估技术成果的关键性能指标。
- ③ 将分析的需求反馈给适用的利益相关方进行审查。
- ④ 识别并解决整套需求中的问题、不足、冲突和弱点。



北京大学

## 2.1 软件生存周期过程

总的来说，成功实施系统或软件的需求定义过程的结果是：

- A) 定义了系统解决方案的系统或元素描述，包括接口、功能和边界。
- B) 定义了系统或软件需求（功能、性能、过程、非功能和接口）和设计约束。
- C) 定义了关键性能度量。
- D) 分析了系统或软件需求。
- E) 系统或软件需求定义所需的任何启用系统或服务都可用。
- F) 开发了系统或软
- G) 件需求与涉众需求的可追溯性。



## 2.1 软件生存周期过程

例如7：验证过程

①为什么要进行验证和确认？

All models are wrong, some models are useful.

--George Edward Pelham Box

出现这一问题的基本原因：

由于人们的认知能力，在抽象一个事物或一个问题时，往往只描述其一些必要性，但不是充分的。例如：“操作系统是一组管理计算机系统资源的程序”。



北京大学

## 2.1 软件生存周期过程

### ②何谓验证过程

是一个确定某项活动的软件产品是否满足在以前的活动中施加于它们的需求和条件的过程。



可见，该过程的目的是：证实一个过程和/或项目的每一软件工作产品和/或服务恰当地反映了已规定的需求。

验证过程可应用于供应、开发、运行或维护等过程。该过程可以由来自同一组织一个人或多个人来实施，也可以由来自另一组织的人员来实施。



北京大学

## 2.1 软件生存周期过程

验证过程包括3个活动：

准备验证      实施验证      管理验证结果

**其中的活动：准备验证**

**主要任务：**

①确定验证策略，考虑验证范围，包括需要验证的软件系统、元素或构件及其预期结果，确定限制验证活动的约束条件，确定优先级。

②从系统或软件需求、架构或设计的验证策略来识别约束条件。

③定义每个验证行动的目的、条件和符合性标准。

④选择适当的验证方法或技术以及验证行动的相关标准，如检查、分析、演示或测试。

⑤确定并计划支持验证所需的必要使能系统或服务。

⑥得到或获取用于支持验证的使能系统或服务的访问权限。

## 2.1 软件生存周期过程

### 其中的活动：管理验证结果

#### 主要任务：

- ①审查验证结果和验证过程中遇到的异常情况，并确定后续行动。
- ②记录验证过程中的事件和问题，并跟踪其解决方案。
- ③获得利益相关方同意，确认软件系统或元素符合规定的要求。
- ④维护已验证软件系统元素的可追溯性。
- ⑤提供纳入基线的关键构件和信息项。



北京大学

## 2.1 软件生存周期过程

总的来说，成功实施验证过程的结果是：

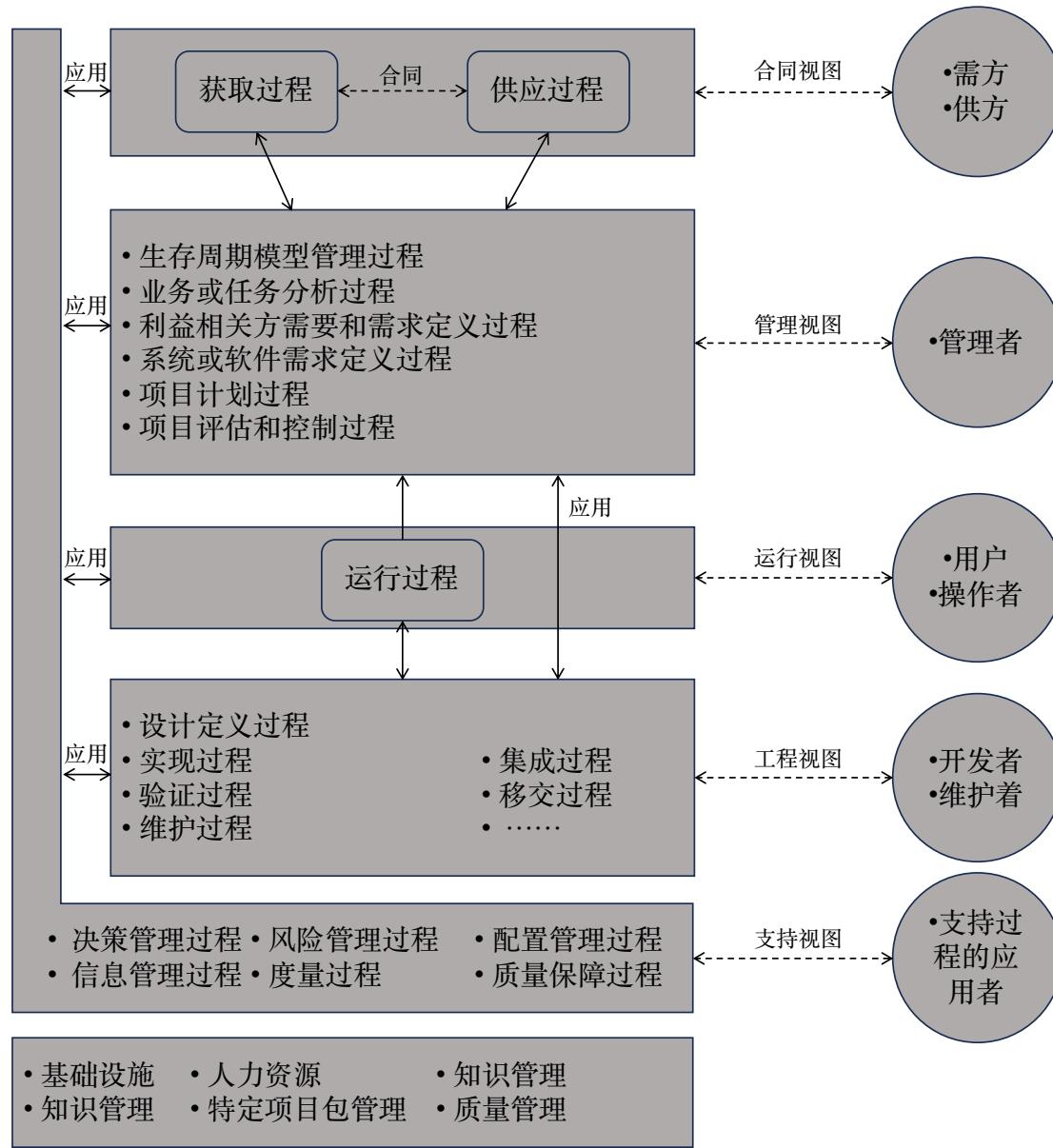
- A) 确定了影响需求、架构或设计的验证约束。
- B) 核查所需的任何使能系统或服务均可用。
- C) 验证了系统或系统元件。
- D) 报告了为纠正措施提供信息的数据。
- E) 提供了实现的系统满足需求、架构和设计的客观证据。



北京大学

# 2.1 软件生存周期过程

## (5) 软件过程之间的关系



北京大学

# 目录

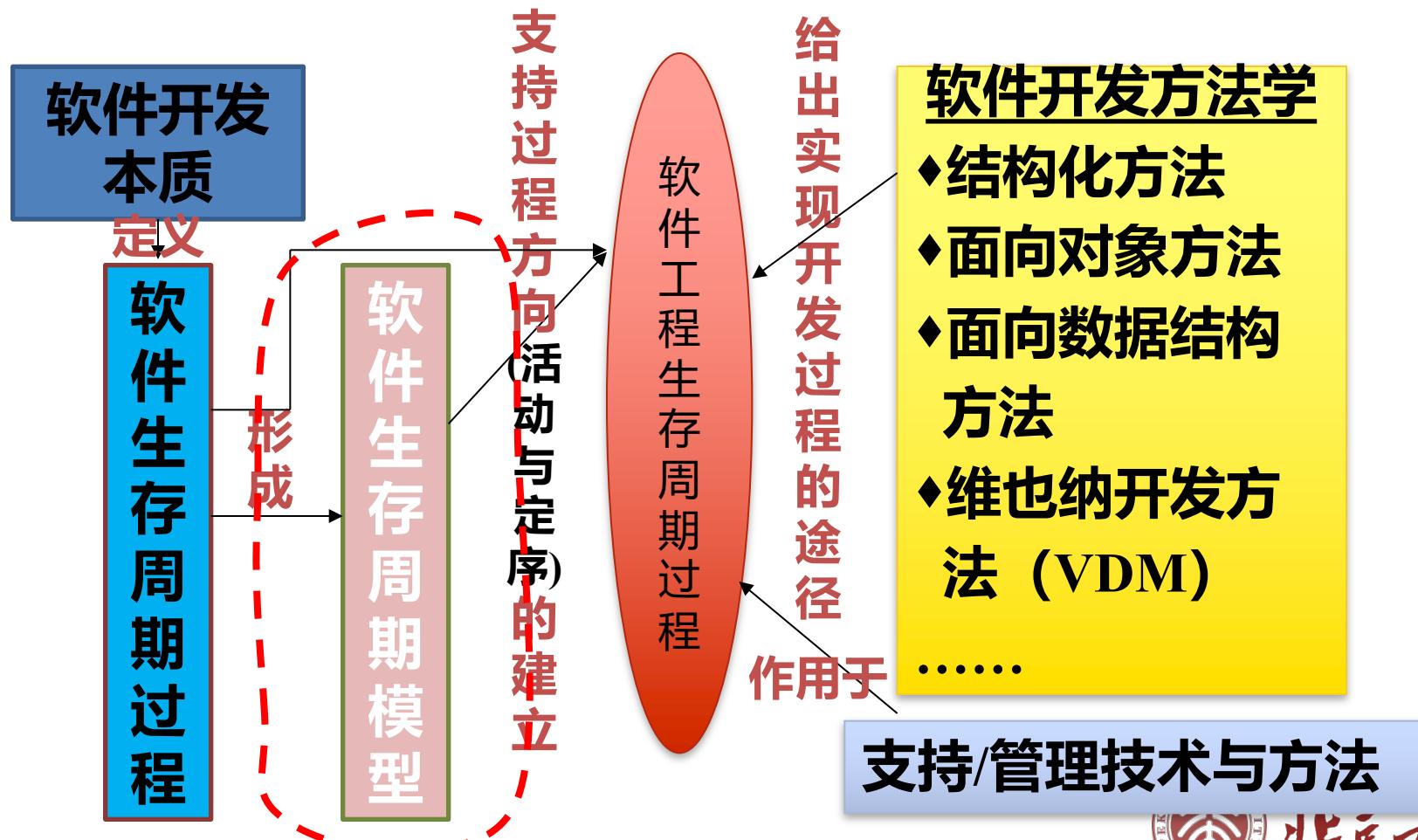
- 2.1 软件生存周期过程
- 2.2 软件生存周期模型
- 2.3 软件过程改进



## 2.2 软件生存周期模型

### 开发活动的组织框架

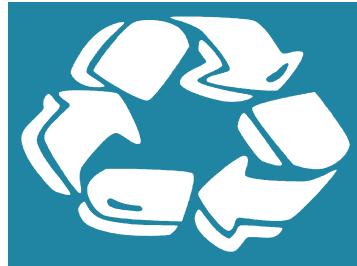
#### -软件生存周期模型



北京大学

## 2.2 软件生存周期模型

### 1、基本概念



**软件生存周期模型 IEEE Standard 12207.0-1996**

软件生存周期模型：一个包括软件产品开发、运行和维护中有关过程、活动和任务的框架，覆盖了从该系统的需求定义到系统的使用终止。

**中国计算机科学与技术百科全书**

称软件生存周期模型为“软件开发模型”，并把它定义为：  
软件过程、活动、任务的结构框架。



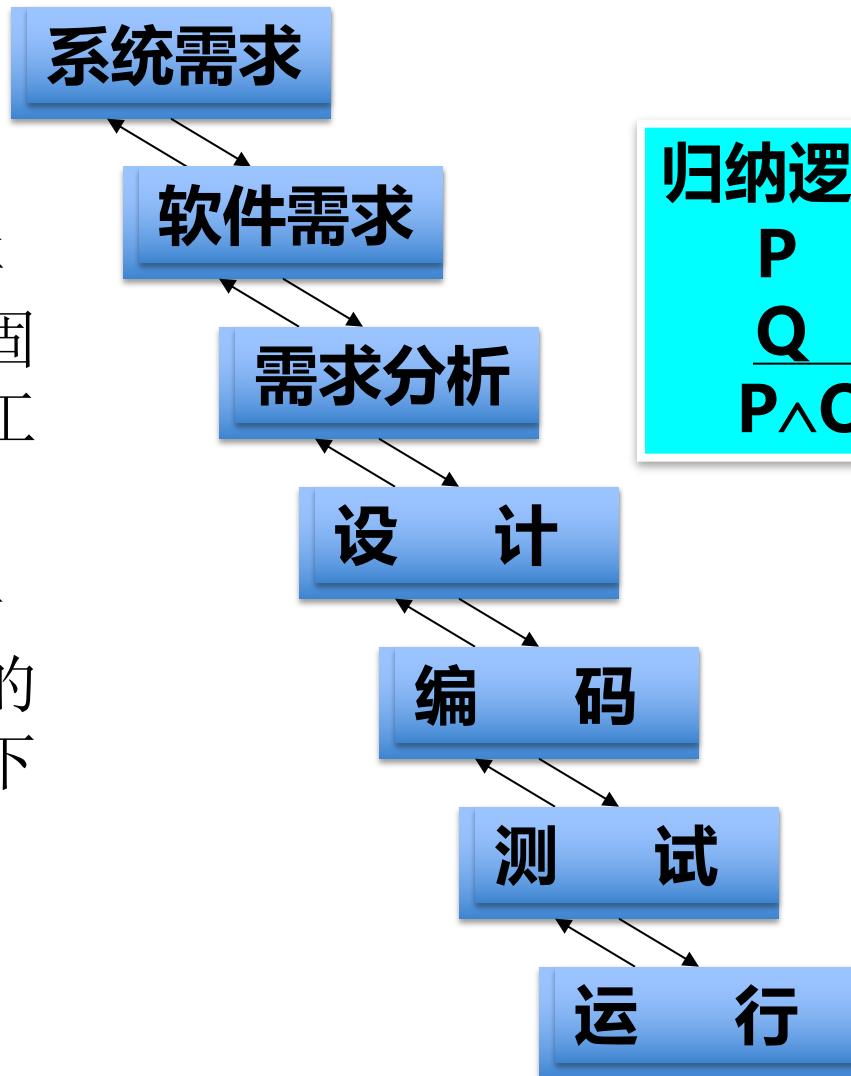
北京大学

## 2.2 软件生存周期模型

### 2、瀑布模型

1970年， W.Royce

- (1) 瀑布模型将软件生存周期的各项活动规定为依固定顺序而连接的若干阶段工作；
- (2) 瀑布模型规定了每一阶段的输入，以及本阶段的工作成果，作为输出传入下一阶段。



北京大学

## 2.2 软件生存周期模型

(1) 项目的开发依次经过：需求、设计、编码和单元测试、集成以及维护—这一基本路径。

(2) 通过每一阶段，提交以下产品：软件需求规约、设计文档、实际代码、测试用例、最终产品等。工作产品（又称可提交的产品，Deliverables）流经“正向”开发的基本步骤路径。

(3) “反向”步骤流表示对前一个可提交产品的重复变更（又称为“返工”(Rework)）。

- 由于所有开发活动的非确定性，因此是否需要重复变更，这仅在下一个阶段或更后的阶段才能认识到。

- 返工不仅在以前阶段的某一地方需要，而且对当前正在进行的工作也是需要的。



北京大学

## 2.2 软件生存周期模型

### 关于瀑布模型的几点说明

#### (1) 瀑布模型的优点

虽然瀑布模型是一个比较“老”的、甚至过时的开发模型，但其优点为：

- ① 在决定系统怎样做之前，存在一个需求阶段，鼓励对系统“做什么”进行规约（即设计之前的规约）。
- ② 在建造构件之前，存在一个设计阶段，鼓励规划系统结构（即编码之前的设计）。
- ③ 在每一阶段结束时进行复审，允许获取方和用户的参与。
- ④ 前一步工作产品可作为下一步被认可的、文档化的基线。允许基线和配置早期接受控制。



北京大学

## 2.2 软件生存周期模型

### (2) 瀑布模型存在的不足

- ①客户必须能够完整、正确和清晰地表达他们的需求；  
开发人员一开始就必须理解需求。
- ②由于需求的不稳定性，使设计、编码和测试阶段都可能发生延期；  
并且当接近项目结束时，出现了大量的集成和测试工作。
- ③缺乏灵活性。一旦软件需求存在偏差，就会导致开发出的软件产品不能满足用户的实际要求。
- ④在一个项目的早期阶段，过分地强调了基线和里程碑处的文档，可能要花费更多的时间，用于建立一些用处不大的文档。
- ⑤在开始的阶段中，很难评估真正的进度状态；并且直到项目结束之前，都不能演示系统的能力，增加了项目的风险。



北京大学

## 2.2 软件生存周期模型

### (3) 瀑布模型适用的情况

- ①需求已被很好地定义和理解；
- ②过程设计人员也很清楚：开发组织非常熟悉为实现这一模型所需要的过程（或经过培训后，熟悉什么时候来支持这一项目，以实现这一模型所需要的过程）。

因此为了避免产生过多的反复迭代工作，增加开发成本，一般在准备采用瀑布模型(也包括其他模型)时，需要考虑以下2个问题：

- (1)过程设计人员必须对初始产品(通常是软件需求规约，SRS)的不确定性进行评估。
- (2)组织是否具有熟练实施每个活动和任务的历史经验。

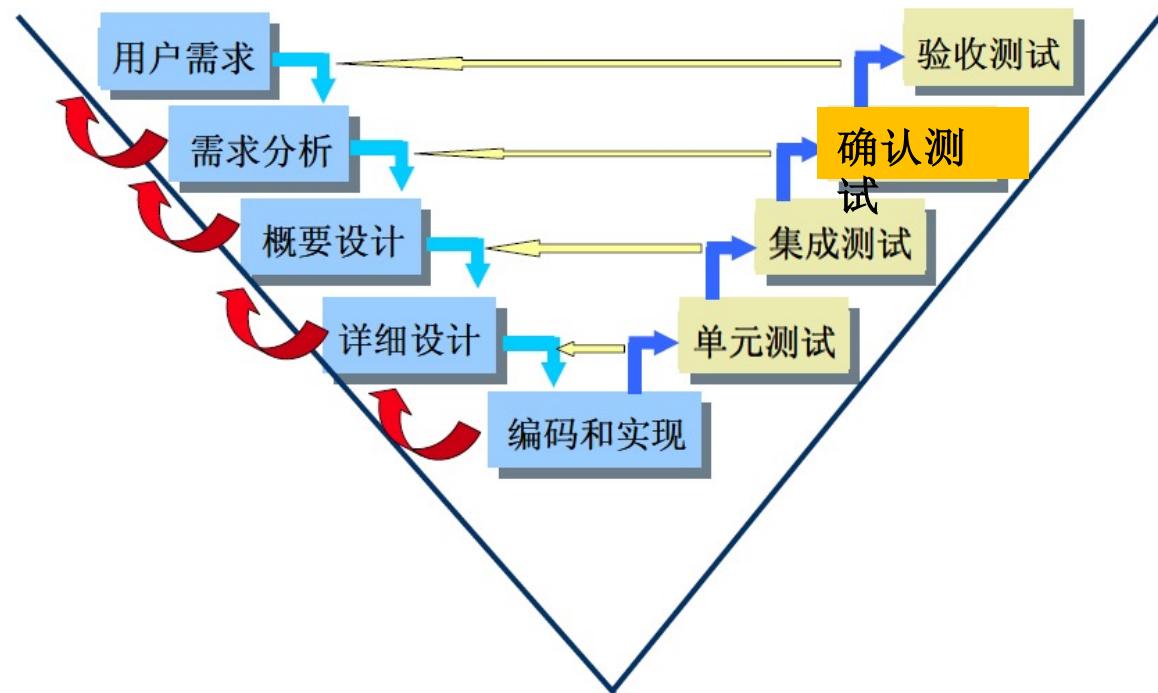


北京大学

## 2.2 软件生存周期模型

### 3、V模型

很多人基于瀑布模型作了改进，尽早发现问题，比较知名的是“V模型”。1978年，Kevin Forsberg 和Harold Mooz提出，应该将系统分解和系统集成的过程通过测试彼此关联，提出了“V模型”。



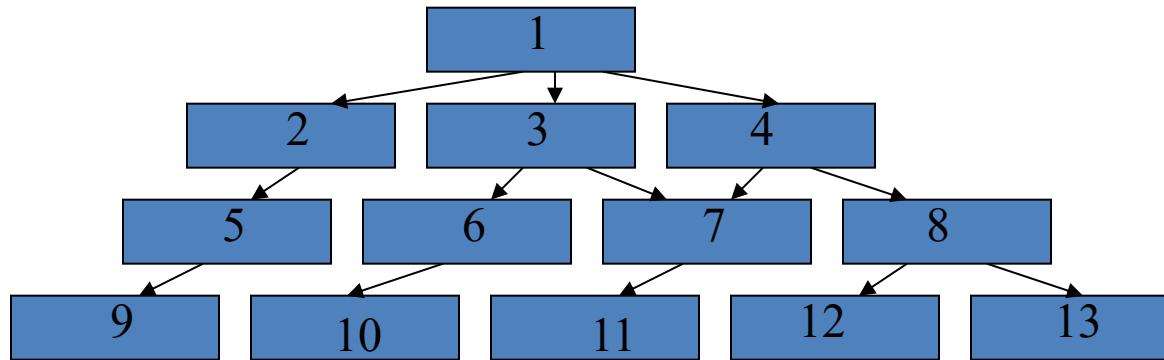
主导思想：研发人员和测试人员需要同时工作，这样尽可能早地找出程序错误和需求偏离，从而更高效地提高程序质量，最大可能地减少成本，同时满足用户的实际软件需求。



北京大学

## 2.2 软件生存周期模型

### 4、增量模型



增量1

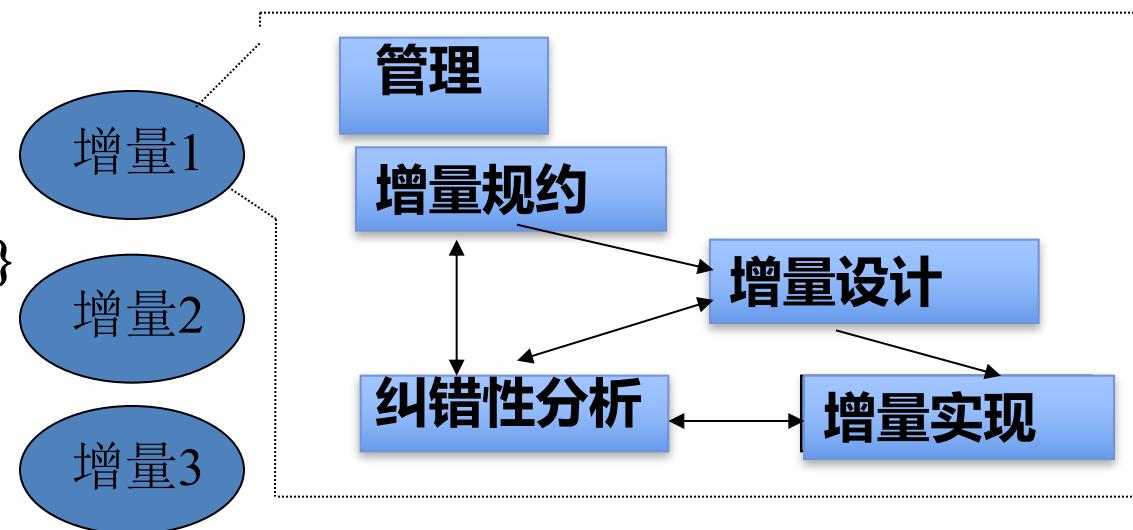
{ 1,2,5,9 }

增量2

{ 3,6,7,4,10,11 }

增量3

{ 8,12,13 }



该模型有一个假设，即需求可以分段，成为一系列增量产品，每一增量可以分别地开发。



北京大学

## 2.2 软件生存周期模型



关于增量模型的几点说明：

### (1) 增量模型的优点

**作为瀑布模型的第一个变体，具有瀑布模型的所有优点。  
此外，它还有以下优点：**

- ①第一个可交付版本所需要的成本和时间是很少的；
- ②开发由增量表示的小系统所承担的风险是不大的；
- ③由于很快发布了第一个版本，因此可以减少用户需求的变更；
- ④允许增量投资，即在项目开始时，可以仅对一个或两个增量投资。

注：如果采用增量投资方式，那么客户就可以对一些增量进行招标。

然后，开发人员按提出的截止期限进行增量开发，这样客户就可以用多个契约来管理组织的资源和成本。



北京大学

## 2.2 软件生存周期模型



(2) 缺点：

**如果增量模型不适于某些项目，或使用有误，则有以下缺点：**

- ①如果没有对用户的变更要求进行规划，那么产生的初始增量可能会造成后来增量的不稳定；
- ②如果需求不像早期思考的那样稳定和完整，那么一些增量就可能需要重新开发，重新发布；
- ③管理发生的成本、进度和配置的复杂性，可能会超出一些组织的能力。



北京大学

## 2.2 软件生存周期模型

### (3) 该模型的适用情况

① 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（构造增量：Build increments. 若一个增量不需要交付给客户的话，那么这样的增量通常称为一个“构造”(Build)。若增量需要被交付的话，那么它们就被认为是发布版本(Released version)。）；

② 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。



北京大学

## 2.2 软件生存周期模型



附：微软“同步-稳定的产品开发模型”

- 将项目分为若干个里程碑阶段
- 定义稳定、灵活的体系结构，并为构件和子系统的开发提供统一的接口
- 开发构件，维持一个可发布的系统版本  
    可以准确把握项目进展情况  
    增强开发人员的信心和成就感  
    可以随时根据市场情况及时作出调整

**Microsoft®**



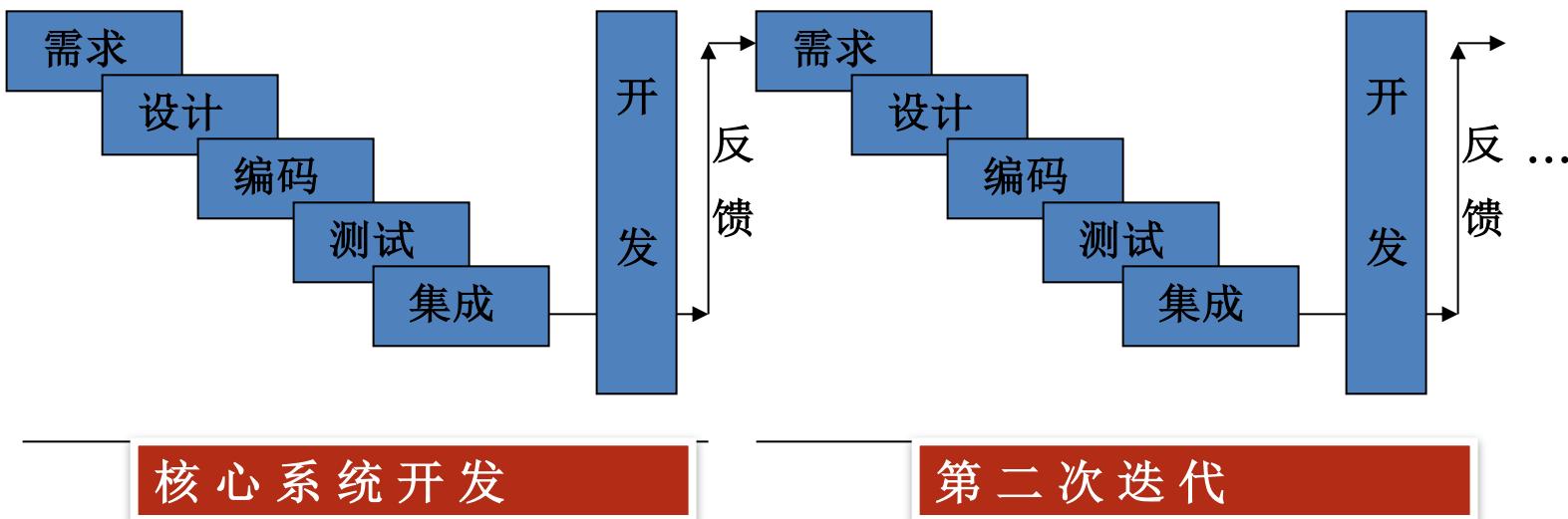
北京大学

## 2.2 软件生存周期模型

### 5、演化模型 (Evolutionary model)

是一种有弹性地过程模式，由一些小的开发步组成，每一步历经需求分析、设计、实现和验证，产生软件产品的一个增量。通过这些迭代，完成最终软件产品的开发。

- 针对事先不能完整地定义需求的软件开发
- 针对用户的核心需求，开发核心系统
- 根据用户的反馈，实施活动的迭代



北京大学

## 2.2 软件生存周期模型

关于演化模型的几点说明

### (1) 主要特征

该模型显式地把增量模型扩展到需求阶段。由图可以看出，为了第二个构造增量，使用了第一个构造增量来精化需求。这一精化可以有多个来源和路径：

首先，如果一个早期的增量已向用户发布，那么用户会以变更要求的方式提出反馈，以支持以后增量的需求开发。

第二，通过实实在在地开发一个构造增量，为以前还没有认识到的问题提供了可见性，以便实际地开始这一增量的工作。



## 2.2 软件生存周期模型

### (2) 与瀑布模型的关系

在演化模型中，仍然可以使用瀑布模型来管理每一个演化的增量。一旦理解了需求，就可以像实现瀑布模型那样开始设计阶段和编码阶段。

### (3) 使用演化模型应注意的问题

不能弱化需求分析阶段的工作。其原因是：在项目开始时，需要考虑所有可能需求源的重要性和风险，并对这些需求源的可用性进行评估，只有这样才能识别和界定不确定的需求，并识别第一个增量中所包含的需求。



北京大学

## 2.2 软件生存周期模型

### (4) 演化模型的长处和不足

其长处与增量模型是类似的,但还具有以下优点:

- ①在需求不能予以规约时, 可以使用这一演化模型。
- ②用户可以通过运行系统的实践, 对需求进行改进。
- ③与瀑布模型相比, 需要更多用户/获取方的参与。

不足有:

- ①演化模型的使用需要有力的管理。
- ②演化模型的使用很容易成为不编写需求或设计文档的借口,  
即使很好地理解了需求或设计。
- ③用户/获取方不易理解演化模型的自然属性, 因此当结果不  
够理想时, 可能产生抱怨。

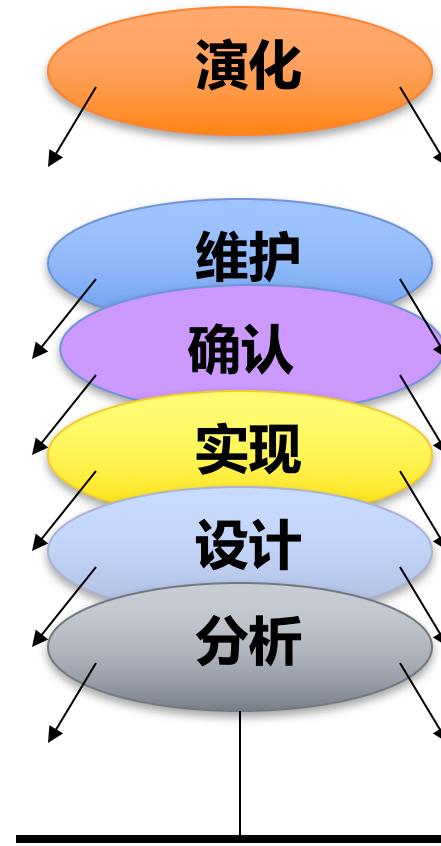


北京大学

## 2.2 软件生存周期模型

### 6、喷泉模型

- 特征：迭代  
无缝
- 与面向对象技术  
的关系



北京大学

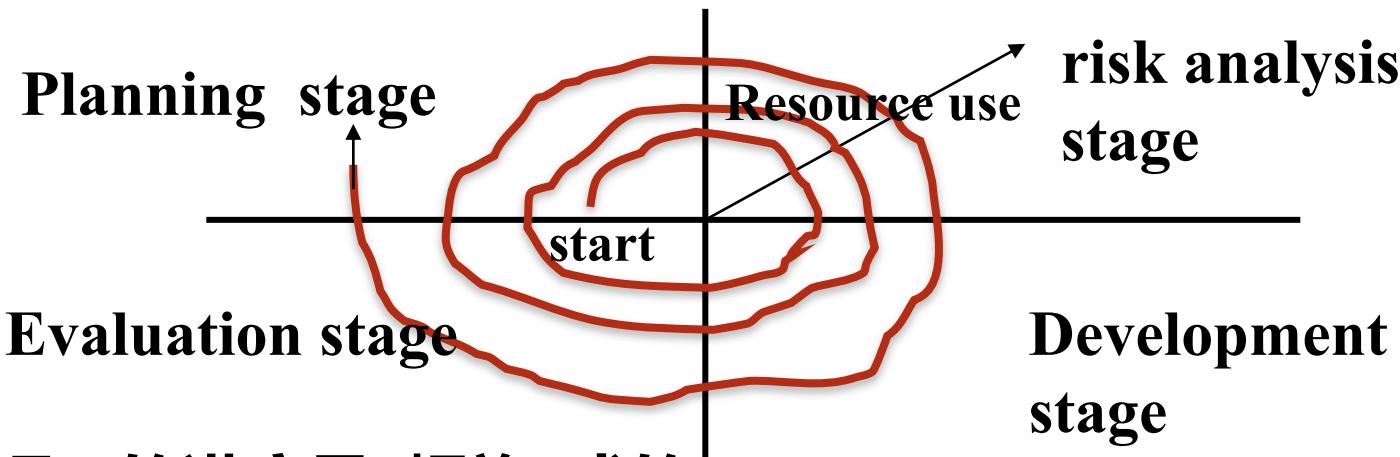
## 2.2 软件生存周期模型

### 7、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：



项目的进度是“螺旋”式的。



北京大学



## 2.2 软件生存周期模型

其中：

- ① 评估和风险分析阶段都可作出一个决策：项目是否继续。
- ② 螺旋循环的次数指示了已消耗的资源；
- ③ 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；
- ④ 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；
- ⑤ 设计和实现活动一般是在开发阶段进行；
- ⑥ V&V 活动在开发阶段和评估阶段进行；



北京大学

## 2.2 软件生存周期模型

### 关于螺旋模型的几点说明：

#### (1) 该模型关注解决问题的基本步骤

- ◆ 标识问题；
- ◆ 标识一些可选方案，选择一个最佳方案；
- ◆ 遵循动作步骤，并实施后续工作。

其中只要完成了开发的一个迭代，开发的另一个迭代就开始。

#### (2) 螺旋模型的一个特征是，实际上只有一个迭代过程真正开发可交付的软件。因此，如果

- ◆ 项目的开发风险很大，或
- ◆ 客户不能确定系统需求，在更广泛的意义上来讲，还包括系统或系统类型的要求，这时螺旋模型就是一个好的生存周期模型。



北京大学

## 2.2 软件生存周期模型

### (3) 与其它模型的关系

①与演化模型一样，螺旋模型也使用瀑布模型作为一个嵌入的过程-即分析、设计、编码、实现和维护的瀑布过程，是螺旋一周的组成部分。



②尽管螺旋模型和一些迭代模型在框架和全局体系结构方面是等同的，但所关注的阶段以及它们的活动是不同的。

具体地说：①标识客户想要的是一个什么样的系统；②确定风险和效益的可选路线；③选择最优方案；④开发系统；⑤评估完成情况等；⑥重新开始。

即：螺旋模型扩展了增量模型的管理任务范围。而增量模型是基于以下假定：需求是最基本的、并且是唯一的风险源。而在螺旋模型中，决策和降低风险的空间是相当广泛的。



北京大学

## 2.2 软件生存周期模型

### 8、原型模型

(1) 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。

(2) 原型在开发中的作用:

- 获得用户的真正需求
- 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

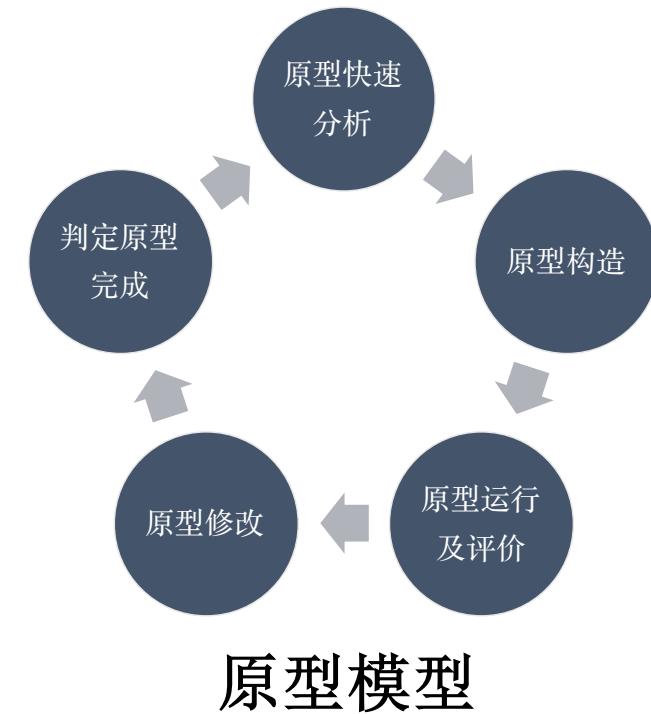
(3) 原型模型中各主要部分:

#### 1) 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。

#### 2) 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。



## 2.2 软件生存周期模型

### 3) 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。

### 4 ) 原型修改

对原型系统，要根据修改意见进行修正。

### 5 ) 判定原型完成

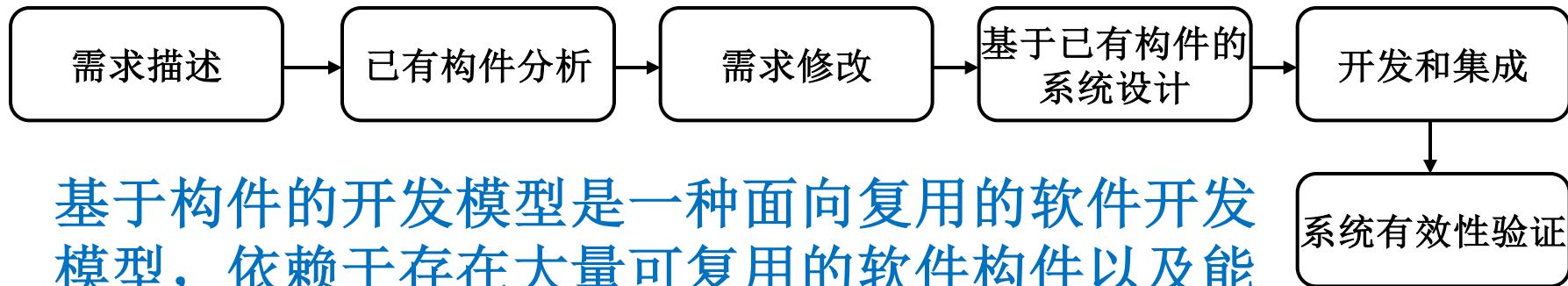
如果原型经过修正或改进，获得了参与者的一致认可，那么原型开发的迭代过程可以结束。



北京大学

## 2.2 软件生存周期模型

### 9、基于构件的开发模型



有三种类型的软件构件可用于基于构件的开发过程：

- 1、通过标准服务开发的Web服务，可用于远程调用。
- 2、对象的集合，作为一个包和构件框架，如.NET或者J2EE等集成在一起。
- 3、独立的软件系统，通过配置在特定的环境下使用。



北京大学

## 2.2 软件生存周期模型

### (1) 基于构件的开发模型的优点

- ①减少了需要开发的软件数量，从而降低了软件开发成本，以及降低了开发中的风险。
- ②可使软件快速地交付。

### (2) 基于构件的开发模型的缺点

- ①需求妥协是不可避免的，这可能导致一个不符合用户真正需要的系统。
- ②对系统进化的控制将失效，因为可复用的构件新版本可能是不受机构控制的。

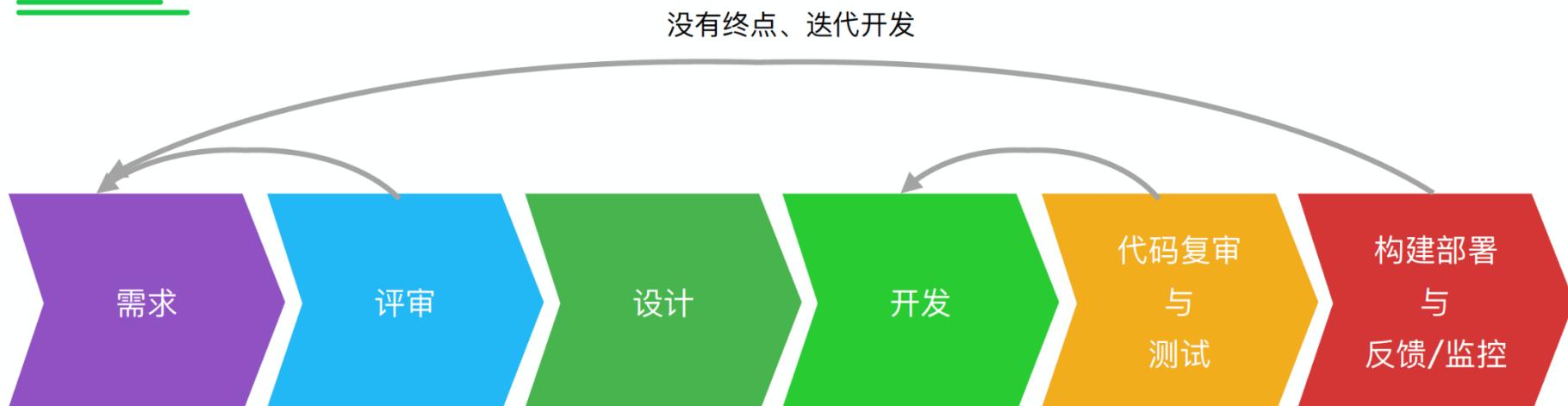


北京大学

## 2.2 软件生存周期模型

### 10、敏捷模型

#### 敏捷模型



参与的需求人员、开发人员、测试人员目标一致，是一个共利益主体。不推崇单向任务传递。



北京大学

## 2.2 软件生存周期模型

- **考虑软件项目的特点**

- 尤其是所开发软件的业务特点，如业务领域是否明确、软件需求是否易于确定、用户需求是否会经常性变化等等
- 是否可以预估到潜在的软件开发风险

- **软件开发团队的水平**

- 需要结合软件开发团队的能力和水平来选择过程模型，以防开发团队和管理人员无法掌控和驾驭过程模型

- **分析软件过程模型特点**

- 优缺点以及适合的场所



北京大学

# 示例：如何选择合适的过程

- **互联网应用软件的软件开发过程模型**

- 特点：软件需求不确定且快速变化
- 如：微信软件，抖音软件，网易云音乐软件等
- 选用瀑布模型不合适，迭代模型、敏捷开发模型较为合适

- **大型航空航天管理软件的软件开发过程模型**

- 特点：软件需求确定且较为稳定
- 如：航天飞行器控制软件
- 可考虑选用瀑布模型和螺旋模型，用敏捷开发模型不是很合适



北京大学

# 目录

- 2.1 软件生存周期过程
- 2.2 软件生存周期模型
- 2.3 软件过程改进



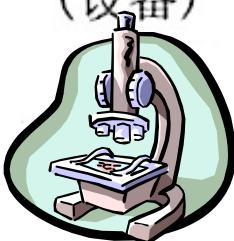
北京大学



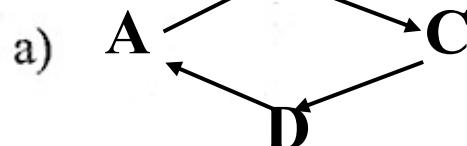
人员

质量  
生产率

技术  
(设备)



过程



乐师及指挥

乐器  
舞台

高水平  
音乐会



b)



## 软件质量三因素及类比



北京大学



Watts Humphrey (CMM项目的主要负责人) 指出：

- 软件系统的质量取决于用来开发和改进它的过程的质量。  
--说明了产品和过程的关系。
- 如果你迷路了，那么即使你有一张地图也是没有用的。  
--说明要进行过程改进，必须对现有的过程有所了解，特别是存在的问题有客观的认识
- 软件过程改进并不是目的地，它只是一个旅程。  
--说明过程改进是一个持续的过程



北京大学

## CMM (the Capability Maturity Model for software)

过程是生产产品的机制。不论是过程改善还是能力确定，均需要过程评估，而过程评估通常基于已提出的一些评估模型。

—— CMM是什么？

—— CMM的知识结构



北京大学

## 一、CMM的发展历史

在80年代中期，美国工业界和政府部门开始认识到：

在软件开发中，关键的问题在于软件开发组织不能很好地定义和控制其软件过程。从而使一些好的开发方法和技术都起不到所期望的作用。

针对这一问题：

- 1986年11月，美国卡内基-梅隆大学软件工程研究所（SEI）开始开发过程成熟度框架，用以促进软件承包商提高产品质量。
- 1987年9月，SEI发布了过程成熟度框架的简要描述和成熟度调查表，用以评价软件承包商的风险。
- 1991年，SEI将过程成熟度框架演化为CMM1.0版：CMU/SEI-91-TR-24、CMU/SEI-91-TR-25。
- 1993年，SEI根据反馈，提出CMM 1.1版：CMU/SEI-93-TR-25。
- 1997年形成了CMM 2.0版，并开始了CMMI的工作。



北京大学

## 二、CMM是什么

- ❖ CMM全称为Capability Maturity Model，即“能力成熟度模型”，是一种描述软件过程关键元素的框架。该模型用以指导软件组织提高其软件开发能力和过程成熟度。它是对于软件组织在定义、实现、度量、控制和改善其软件过程的实践中各个发展阶段的描述。
- ❖ CMM指的是软件过程能力成熟度模型，按软件过程的不同成熟度划分了5个等级，1级被认为成熟度最低，5级成熟度最高。
- ❖ CMM给出了从混乱、个人的过程到成熟的规范化过程的一个框架。
- ❖ CMM项目的主要负责人Mark Paultk指出：“软件组织可以通过CMM去定义、实施、度量、控制和改进自己的软件过程”。人们可以利用该框架进行可靠且统一的评估，实现对软件过程的度量。
- ❖ CMM体现了软件工程和软件管理的优秀实践
- ❖ CMM不是银弹，它并没有涉及是的项目成功的所有重要问题。



北京大学

### 三、CMM族

自从CMM出现以来，出现了多种基于CMM的模型。例如，  
人力资源能力成熟度模型、系统工程能力成熟度模型等等：

- (1) SW-CMM (Software CMM) 软件CMM
- (2) SE-CMM (System Engineering CMM) 系统工程CMM
- (3) SA-CMM (Software Acquisition CMM) 软件采购  
CMM
- (4) IPD-CMM (Integrated Product Development CMM)  
集成产品开发CMM
- (5) P-CMM (People CMM) 人力资源能力成熟度模型

为了以示区别，国内外很多资料把CMM叫做SW-CMM。



北京大学

## 四、CMMI的发展

SEI在SW-CMM2.0版本(这个版本并没有发布)形成过程中，又启动了一个新的项目，称为**CMMI**（**Capability Maturity Model Integration**），即集成的能力成熟度模型，是由美国国防部与卡内基-梅隆大学和美国国防工业协会共同开发和研制的系统工程和软件工程的集成成熟度模型。

- CMMI的产生是为了将各种能力成熟度模型集成起来：
  - 软件能力成熟度模型（**SW-CMM**） 、
  - 系统工程能力成熟度模型（**SE-CMM**） 、
  - 集成产品开发能力成熟度模型（**IPD-CMM**）
  - 软件采购能力成熟度模型（**SA-CMM**）



北京大学

# 五、CMMI的描述形式

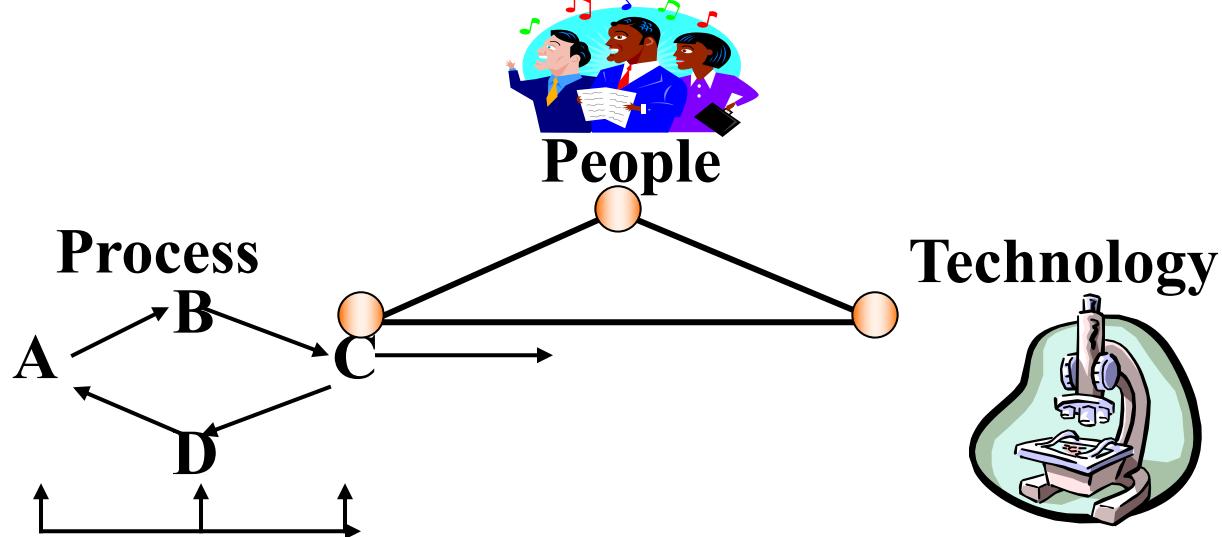
- CMMI在CMM的基础上提供了两种模型描述形式：阶梯式和连续式
  - 阶梯式的描述方式与CMM类似，通过将过程改进划分为5个等级和若干个过程域，逐步实现软件成熟度的从低到高发展。这个过程中，每一阶段都是下一个阶段的基础。
  - 连续式的描述方法则允许组织选择过程改进顺序，给组织在进行过程改进的时候带来更大的自主性，不用再像CMM模型一样受到严格的等级限制。例如在阶梯式描述方法下，组织必须先实现项目管理的所有关键过程域，并达到特定的等级，才能进入下一阶段。
    - 在连续式描述方法下，组织可以选择自己的改进顺序，并在其自己的时间表上逐步实现相应的目标和实践。
    - 这样的改进增加了灵活性，但由于缺乏指导，一个组织可能缺乏对关键过程域之间依赖关系的正确理解而片面的实施过程，从而使部分软件过程成为空中楼阁，缺乏其他过程的支撑。
- CMMI的过程域与国际标准化组织和国际电子技术协会 **ISO/IEC 15504** 也非常相似，因此可以提供过程改进与 **ISO/IEC15504** 之间的简单比较。



## 六、CMM和CMMI的基本内容

### 1、基本思想

支撑软件产品 / 系统质量的三大要素：



“整个软件任务可以看作是一个过程，该过程可以予以控制、测量和改进”

-----Watts S.Humphrey



北京大学

## 2、基本概念：

### (1) 过程

过程（Process）是一种手段，通过该手段可以把人、规程、方法、设备以及工具进行集成，以产生一种所期望的结果。

### (2) 过程能力

① 定义：(开发组织或项目组)通过遵循其软件过程能够实现预期结果的程度。

一个组织的软件过程能力，是未来项目结果的指示器，给出了一种预测该组织承担下一个软件项目可能结果的方法。是不同等级过程能力的基本指标。



## ②低过程能力的基本特征

- ▶ 非常依赖当前的参与人员（practitioners）；
- ▶ 软件过程与管理均是临时准备；
- ▶ 没有严格的下一步；                   ▶ 冒险地使用新技术；
- ▶ 复审和测试常常不足；                ▶ 产品质量很难预测
- ▶ 交付的“东西”不符合要求；        ▶ 进度延迟和预算超额。

## ③高过程能力的特征

- ▶ 定义了过程，建立了使用技术的基础；
- ▶ 开发和管理遵循一个确定的途径；
- ▶ 过程得到了很好的控制，并得到各方面（包括测量）的支持；
- ▶ 实现了过程制度化，并不断改进。

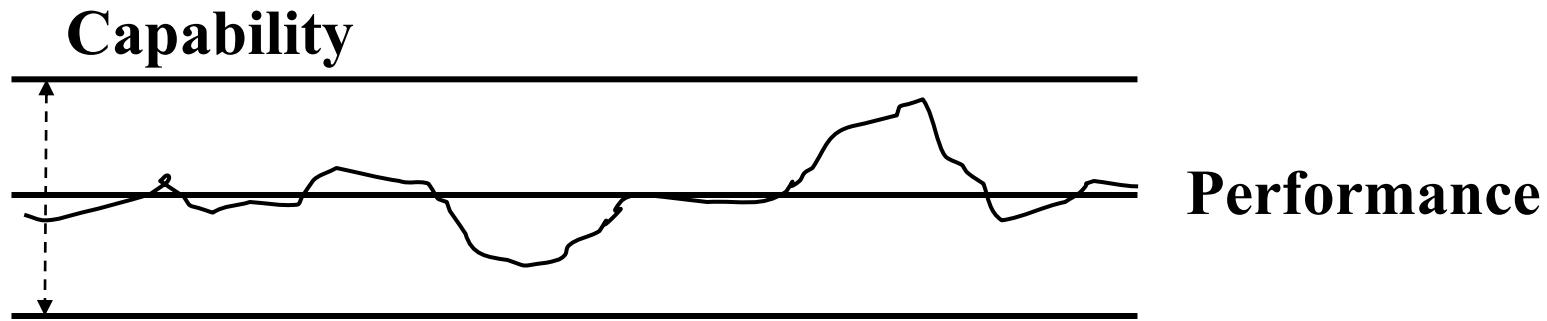


北京大学

### (3) 过程性能 (Process Performance)

遵循一个过程所达到实际结果的一个测度 (measure)

过程能力和过程性能之间的关系



注意：软件过程能力与软件过程性能之间的关系：

- ① 一个是能够实现预期结果的程度，一个是得到的实际结果
- ② 一个项目的实际过程性能，可能并不充分反映其所在组织的整个过程能力。（由于该项目的具体属性和执行该项目的环境所限）

## (4) 过程成熟度 ( Process Mature )

一个特定软件过程被明确和有效地定义、管理、测量和控制的程度。

( Definition: Process Mature

The extent to which a specific process is explicitly:

- ① defined(you know what is done),
- ② managed (you can control the process qualitatively ),
- ③ measured(you know how much is done, and how well),
- ④ controlled(you can control the process quantitatively),
- ⑤ effective(you can improve the process rationally) )



北京大学

## 软件过程成熟度指明：

- 一个软件开发组织**软件过程能力**的增长潜力；  
    --能力提高的**基础性**
- 表明一个开发组织**软件过程的丰富多样性**，  
    --能力提高的**可能性**
- 在各开发项目中运用**软件过程的一致性**。  
    --能力提高的**持续性**

这意味着：由于开发组织通过运用软件过程，使各项  
目执行软件过程的纪律性一致地增强，导致软件生产率和  
质量可以得到不断地改进。



北京大学

## 组织成熟度（Organizational Maturity）

- 组织的成熟度是由一组过程的组合能力来表达的，其中包括支持它们的制度因素（factor）
  - 高的组织成熟度，是将组织的一组过程看作为一个整体，该整体是高的过程能力。其主要表现为：
    - ◆ 不论是开发还是管理，均有明确、严格的途径；
    - ◆ 定义了组织过程并不断改善之；
    - ◆ 得到了管理人员和其他人员的支持；
    - ◆ 实施了很好的控制；
- .....



北京大学

## (5)能力成熟度等级

- CMM的核心是把软件开发视为一个过程，并根据这一原则对软件开发和维护进行过程监控和研究，使其更加科学化、标准化、使企业能够更好地实现商业目标。
- CMM框架使用5个等级来评价软件过程能力

初始级（等级1）是混乱的软件过程

可重复级（等级2）是经过训练、有纪律的软件过程

已定义级（等级3）是标准一致的软件过程

已管理级（等级4）是可预测的软件过程，

优化级（等级5）是可持续改进的软件过程

CMM还为每一个成熟度等级规定了一些目标和与之相关的若干个关键过程域（KPA），过程除了需要达到成熟度等级，还需满足每个成熟度等级的关键过程域。

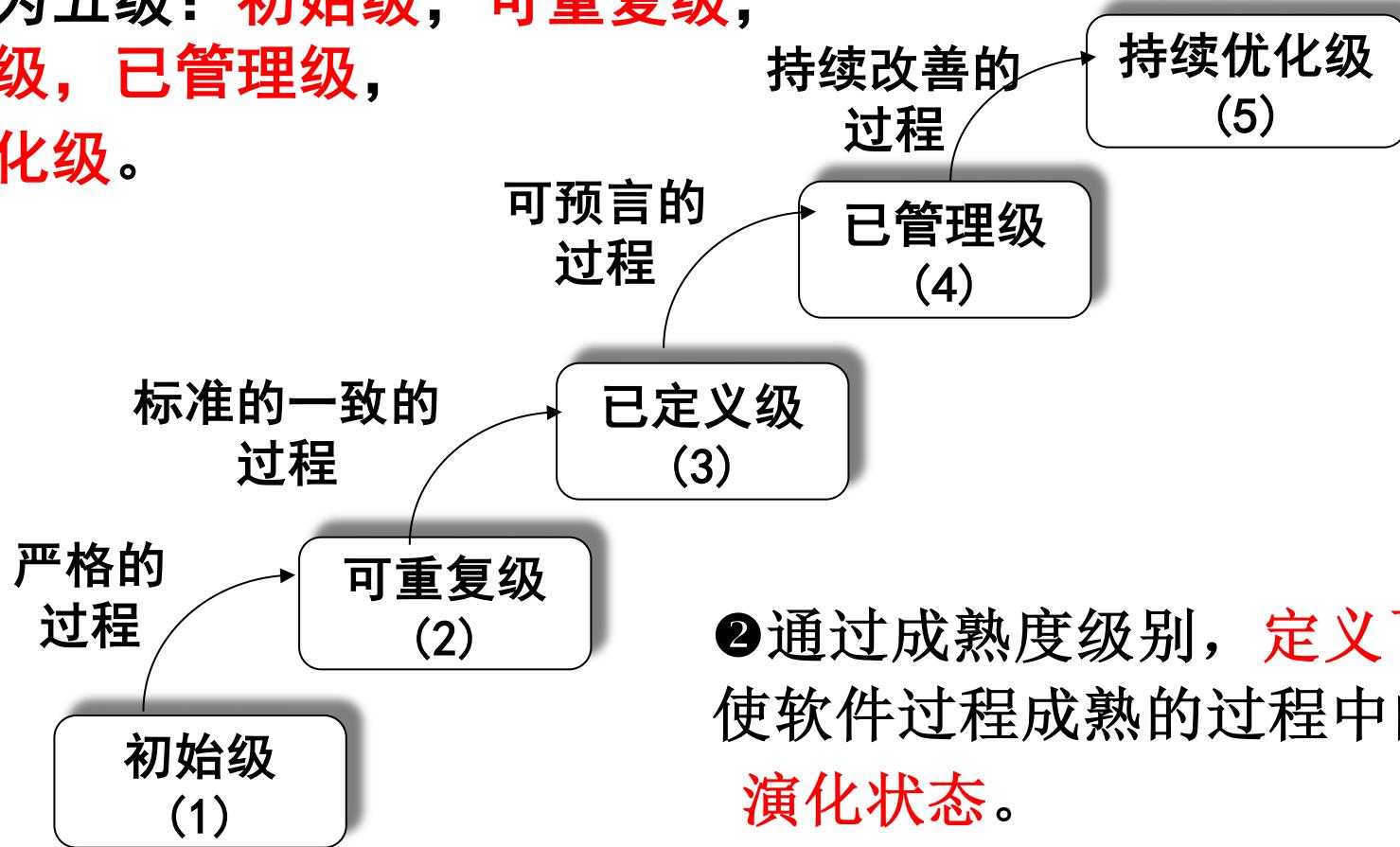


北京大学

### 3、CMM的软件过程成熟度框架

#### (1) 成熟度框架

①在这一框架中，将过程能力成熟度分为五级：**初始级，可重复级，已定义级，已管理级，持续优化级。**



②通过成熟度级别，**定义了**在使软件过程成熟的过程中的**演化状态**。



北京大学

- CMM的核心是把软件开发视为一个过程，并根据这一原则对软件开发和维护进行过程监控和研究，使其更加科学化、标准化、使企业能够更好地实现商业目标。
- CMM框架使用5个等级来评价软件过程能力：

初始级（等级1）是混乱的软件过程

可重复级（等级2）是经过训练、有纪律的软件过程

已定义级（等级3）是标准一致的软件过程

已管理级（等级4）是可预测的软件过程，

优化级（等级5）是可持续改进的软件过程

**CMM还为每一个成熟度等级规定了一些目标和与之相关的若干个关键过程域（KPA），过程除了需要达到成熟度等级，还需满足每个成熟度等级的关键过程域。**



## 过程成熟度框架：

描述：一条从无序的、混乱的过程达到成熟的、有纪律的软件过程的进化途径。

用途：以软件过程成熟度框架，可以导出过程改进策略，为软件过程的不断改进的历程提供了一份导引图：  
--指导软件开发组织不断识别出其软件过程的缺陷  
--引导开发组织在各个等级“做什么”改进（但它并不提供“如何做”的具体措施。

基础：软件过程成熟度框架的基础是等级内部结构。

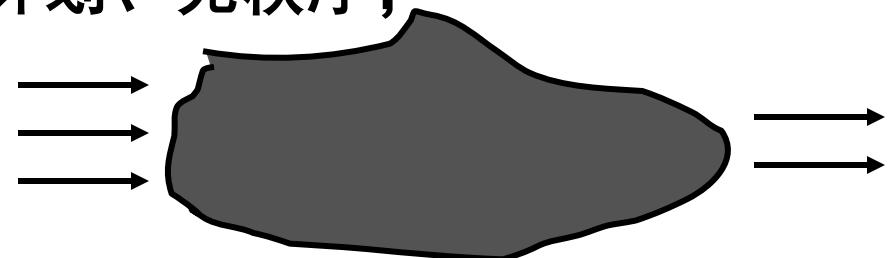


北京大学

## (2) 各等级的基本特征

- **初始级** 主要特征：

- 组织：组织通常没有提供开发和维护软件的稳定的环境。
- 项目：当发生危机时，项目通常放弃计划的过程，回复到编码和测试。
- **过程能力：不可预测。(unpredictable)** 由于：
  - ① 软件开发无规范；
  - ② 软件过程不确定、无计划、无秩序；
  - ③ 过程执行不“透明”；
  - ④ 需求和进度失控。



**结果：**项目的成败完全取决于个人的能力和努力；

软件性能随个人具有的技能、知识和动机的不同而变化，  
并只能通过个人的能力进行预测。

- 可重复级

- 主要特征：

- 组织：将软件项目的有效管理过程制度化，这使得组织能够重复以前项目中的成功实践。

- 项目：配备了基本的软件管理控制。

- 过程能力：

- ① 可重复的：即对当前项目的需求分析后制定的，能重复以前的成功实践，尽管在具体过程中可能有所不同。  
这是该级的一个显著特征



北京大学

②基本可控的：即对软件项目的管理过程是制度化的。

- 在项目的规划和服务跟踪过程中规定并设置了监测点；
- 对软件需求和为实现需求所开发的软件产品建立了基线；
- 为管理、跟踪软件项目的成本、进度和功能提供了规范；
- 提供了当不满足约定时的识别方法和纠偏措施。

软件项目过程基本上是可视的

③过程是有效的：即对项目建立了实用的、已文档化的、已实施的、已培训的、已测量的和能改进的过程。

项目的过程基本是可特征化的



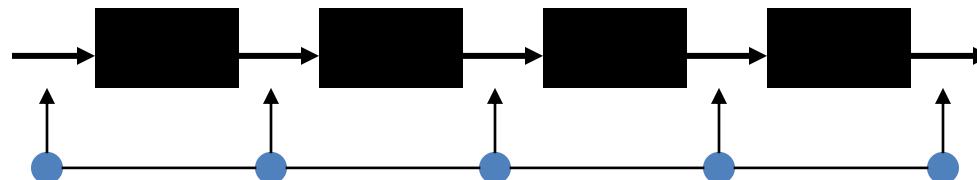
北京大学

④项目是稳定的：即对新项目的策划和管理，有明确的管理方针和确定的标准（包括对分承制方），可使项目的进展稳定。

新项目的策划和管理是基于成功项目经验的

⑤过程是有纪律的：即对所建立和实施的方针、规程，对软件项目过程而言，已进化为组织的行为。从而使软件开发组织能够保证准确地执行给定的软件过程。

总之，2级的过程是可视的，即可以获取项目运行状态。



北京大学

□ 具备以上过程能力特征的途径:

➤ 实现关键过程域:

软件配置管理、软件质量保证、软件子合同管理、  
软件项目跟踪和监督、软件项目规划、需求管理。

其中:

- 过程域:互相关联的若干个软件实践活动和有关基础设施的集合, 即《软件活动, 基础设施》。
- 关键过程域:对某一成熟度等级将起到至关重要的过程域即它们的实施将对达到该成熟度等级的目标起保证作用, 这些过程域被称为关键过程域。
- 每一软件过程成熟度等级均包含一组特定的关键过程域。



- **已定义级**
- 实现了可重复级（2级）的关键过程域：
  - 软件配置管理、软件质量保证、软件子合同管理、
  - 软件项目跟踪和监督、软件项目规划以及需求管理
- 实现了关键过程域：
  - 组织过程焦点、组织过程定义、培训大纲、集成软件管理、软件产品工程、组间协调以及同行评审
- 主要特征：
  - 组织：在组织范围内开发和维护软件的标准过程被文档化，其中包括软件工程过程和管理过程，它们集成为一个一致的整体。
  - 项目：对组织的标准软件过程进行裁剪，来开发它们自己项目的软件过程。

**过程能力：是标准的和一致的。 (standard and consistent)**

① 建立了“组织的标准软件过程”：即

- 关注的焦点转向组织的体系和管理；
- 全组织建立了软件开发和维护的标准过程；
- 软件工程过程和软件管理过程，被综合为一个有机的整体，并且已经**文档化**。

② 建立了负责组织的软件过程活动的机构：即

在软件组织中存在负责软件过程活动的机构，并具体实施全组织的过程制定、维护和改进。

其中包括全组织的人员培训，使之具备必须的技能和知识，能高效地履行其职责。



### ③ 项目定义的软件过程：即

项目能够依据其环境和需求等，通过剪裁组织的标准过程，使用组织的过程财富，自定义项目的软件过程。其中，允许有一定的自由度，但任务间的不匹配情况，应在过程规划阶段得到标识，并进行组间协调和控制。

### ④ 组织可视项目的进展：即

由于项目自定义的软件过程将开发活动和管理活动综合为一个协调的、合理定义的软件过程，并明确规定了每一活动的输入、输出、标准、规程和验证判据。

因此：

管理者或软件项目负责人能够洞察

所有项目的技术进展、费用和进度



⑤组织的软件能力均衡、一致：即

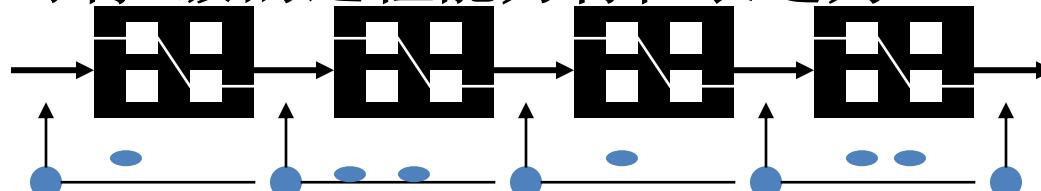
由于：

- 整个组织范围内的软件开发和维护过程已经标准化；
- 软件工程技术活动和软件管理活动都实现文档化的规范管理；
- 组织和项目的软件过程都是稳定的、和可重复的；
- 这种过程能力是建立在整个组织范围内对已定义过程中的活动、作用和职责的共同理解基础之上。

因此：

在整个组织范围内软件能力是均衡、一致的

综上，可将3级的过程能力特征表述为：



北京大学

## •已管理级

□实现了关键过程域：定量过程管理和软件质量管理。

□主要特征：

—项目：项目减小过程性能的变化性，使其进入可接收的量化边界，从而达到对产品和过程的控制。

—组织：为软件产品和过程都设定了量化的质量目标。

—过程能力：可预言的。**(predictable)**

① 设置了定量的质量目标：即

- 组织对软件产品和过程设置了定量的质量目标；
- 软件过程具有明确定义和一致的测量方法与手段；  
可以定量地评价项目的软件过程和产品质量



北京大学

②项目产品质量和过程是受控和稳定的：即  
可以将项目的过程性能变化  
限制在一个定量的、可接受的范围之内。

产品质量和过程是受控和稳定的

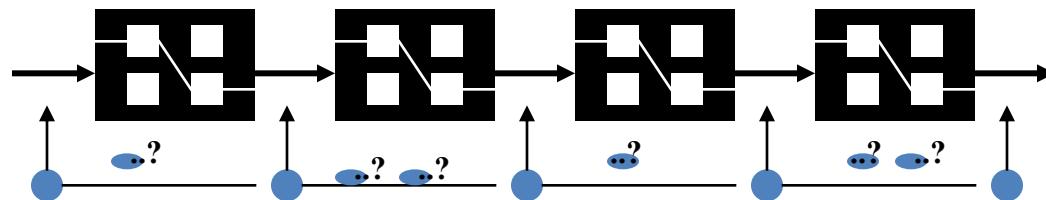
③开发新领域软件的风险是可定量估计的：即  
由于组织的软件过程能力是已知的，从而可以利用全  
组织的软件过程数据库，分析并定量地估计出开发新领域软  
件的风险。



北京大学

④ 组织的软件过程能力是可定量预测的：即  
过程是经测量的并能在可预测的范围内运行，一旦发现  
过程和产品质量偏离所限制的范围时，能够立即采取措  
施予以纠正。

综上，该级的过程能力特征可表述为：



北京大学

- **持续优化级**

- 实现了关键过程域：

- 缺陷预防、技术变化管理、过程变化管理

- 主要特征：

- 组织：关注于持续的过程改进。
  - 项目：软件过程被评价，以防止过失重复发生，从中获得的教训散布给其它项目。
  - **过程能力：持续的改善。 (continuously improving)**

- (1)过程不断改进**，即组织注重不断地进行过程改进。

- 组织有办法识别出过程的弱点，并及时地予以克服；
    - 能够利用关于软件过程有效性的数据，识别最佳软件工程实践的技术创新，并推广到整个组织。



北京大学

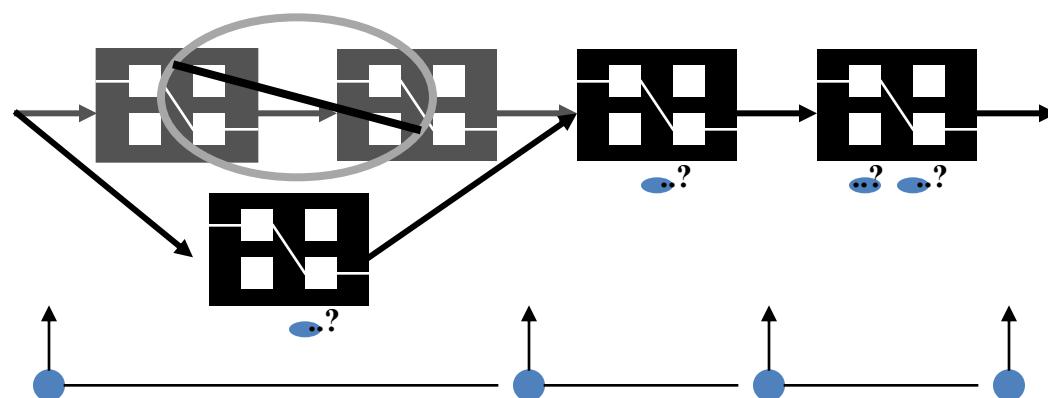
## (2) 缺陷能有效预防: 即

软件项目组能分析并确定缺陷的发生原因, 认真评价软件过程, 以防止同类缺陷再现, 并且能将经验告知其他项目组。

## (3) 组织的过程能力不断提高: 即

组织既能在现有过程的基础上以渐进的方式, 又能以技术创新等手段, 不断努力地改善过程性能。

综上, 该级的过程能力特征可表述为:



北京大学

## 关于五个级别的3点说明

- ① 从第1级提升到第2级可能需要几年的时间，在其它级别间提升通常依次需要2年的时间。
- ② 第1级组织的成功依赖于组织中人员的能力。对于所有成熟度级别的组织来说，选择、雇用、培养和保持有能力的人员都是重要的问题，但这超出了CMM的范围。
- ③ 每个级别为以后的级别有效地和有效率地实现过程提供基础。跳过级别是达不到预期的目标的。

例如1：第1级的组织，如果在建立可重复的过程(第2级)之前，试图实现定义的过程(第3级)，通常是不会成功，因为项目管理者被进度和开销压力所淹没。



北京大学

**例如2:组织在没有定义过程的基础时,如果试图实现管理的过程(第4级),通常是不会成功的。**  
**因为没有定义过程,就没有解释度量的共同基础。**

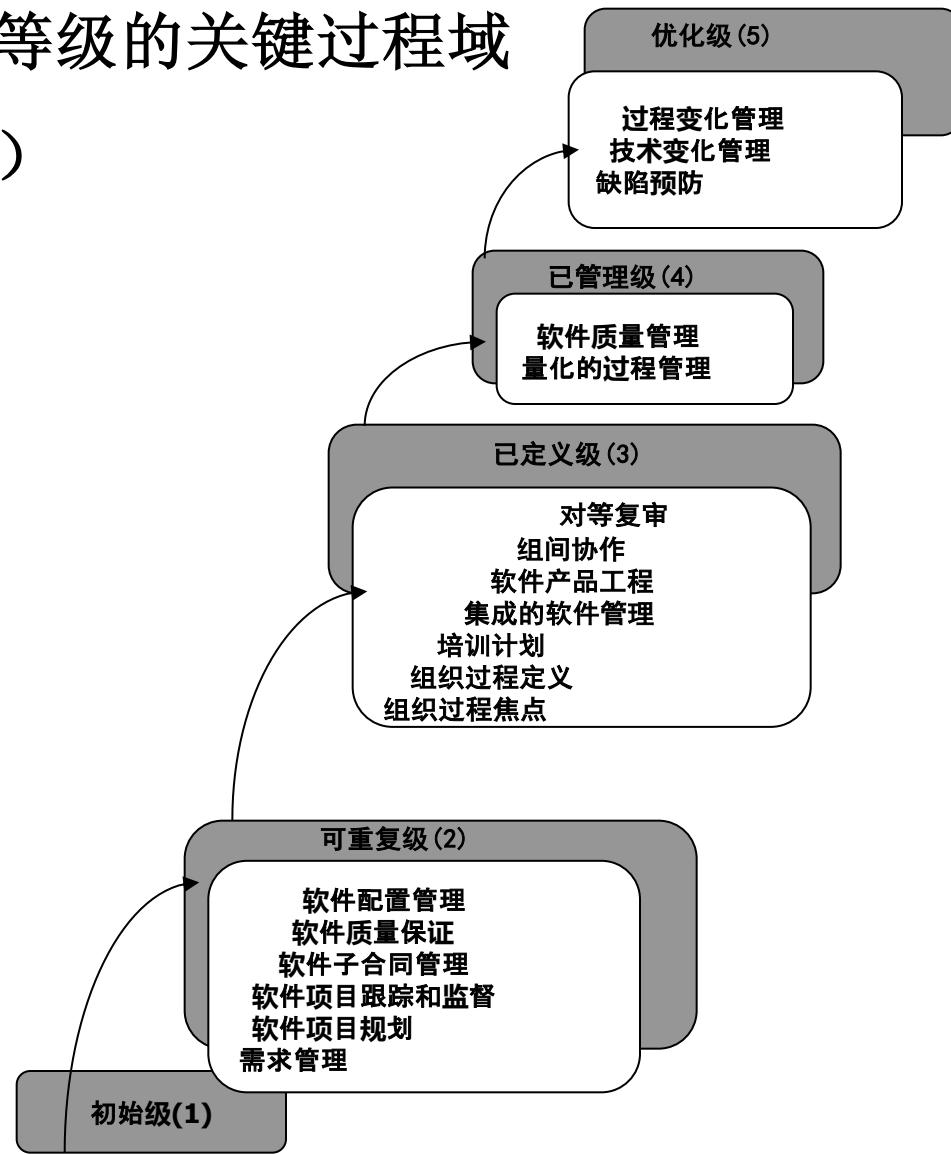
**例如3:组织在没有管理过程(第4级)的基础时,如果试图实现优化过程(第5级),通常会失败的。**  
**因为对过程改变的影响缺乏理解。**

**注意:** ② 和③两点,说明了成熟度等级的演化特性。



北京大学

# 汇总:CMM各等级的关键过程域 (共18个)



北京大学

# 4、CMMI的过程成熟度框架

CMMI将能力成熟度

- 分为5个等级：

初始级、已管理级、已定义  
级、量化管理级、优化级

- 22个关键过程域

CMMI等级	过程域中文名称	过程域英文名称	过程类型
第2级 已管理级 7个过程域	需求管理	Requirements Management	工程
	项目规划	Project Planning	项目管理
	项目监控	Project Monitoring and Control	项目管理
	供应商协议管理	Supplier Agreement Management	项目管理
	度量分析	Measurement and Analysis	支持
	过程和产品质量保证	Process and Product Quality Assurance	支持
	配置管理	Configuration Management	支持
第3级 已定义级 11个过程域	需求开发	Requirements Development	工程
	技术方案	Technical Solution	工程
	产品集成	Product Integration	工程
	验证	Verification	工程
	确认	Validation	工程
	组织过程焦点	Organizational Process Focus	过程管理
	组织过程定义	Organizational Process Definition	过程管理
	组织培训	Organizational Training	过程管理
	集成化项目管理	Integrated Project Management	项目管理
	风险管理	Risk Management	项目管理
	决策分析与解决方案	Decision Analysis and Resolution	支持
第4级 量化管理级 2个过程域	组织过程绩效	Organizational Process Performance	过程管理
	定量项目管理	Quantitative Project Management	项目管理
第5级 优化级 2个过程域	组织革新与推广	Organizational Innovation and Deployment	过程管理
	原因分析与解决方案	Causal Analysis and Resolution	支持

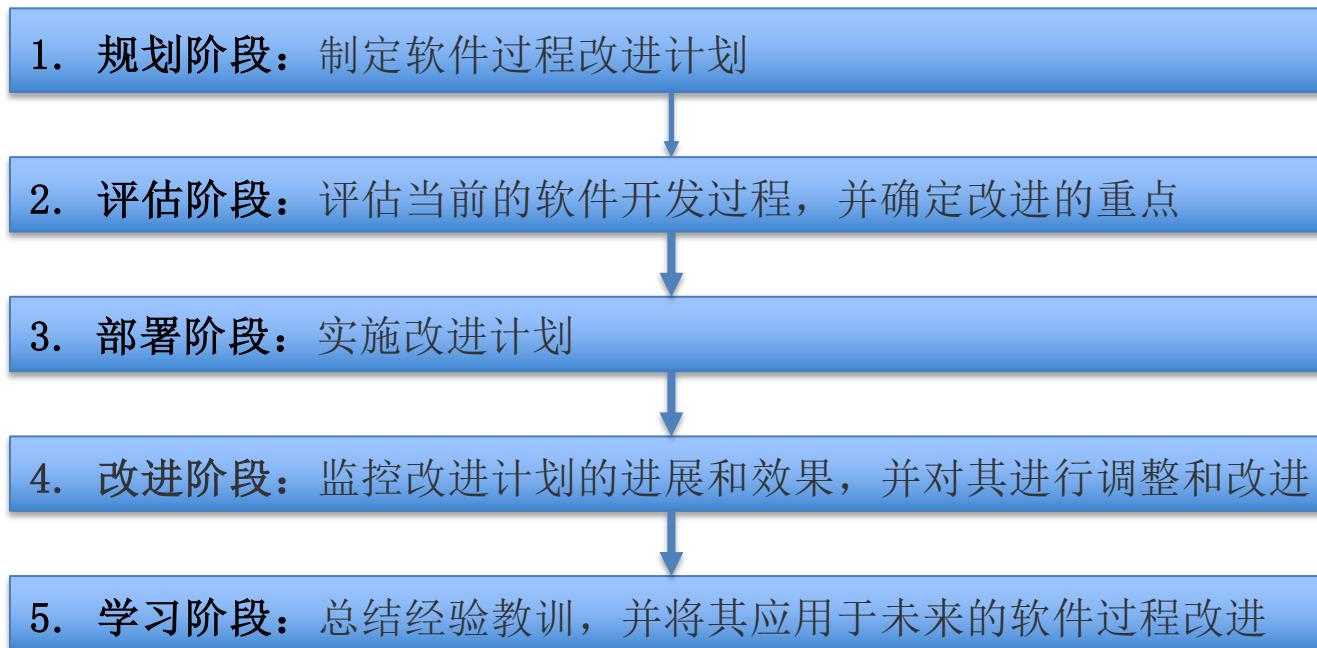
## 5、CMM和CMMI的等级及其基本特征

- 总的来说，CMM和CMMI都是软件工程领域最常用的软件过程成熟度模型。这两个模型的目标都是提高组织的软件工程过程能力，以提高产品质量和项目成功率。它们都使用了5个等级来表示其成熟度等级，但在5个等级的划分和每个等级内的关键过程域方面略有不同。
- CMM模型的5个等级分别是初始级、重复级、定义级、管理级和优化级。初始级和重复级主要关注项目管理和过程管理的基础建设，定义级和管理级则关注于建立和执行规范化的软件开发和维护过程。而优化级则强调持续过程改进和创新，关注于如何不断提高过程能力和性能。**CMMI模型的5个等级分别是初始级、管理级、定义级、量化管理级和优化级。**它们的定义和关注点与CMM模型类似，但CMMI模型在每个等级内的关键过程域方面更加具体和详细。CMMI模型还涵盖了软件开发和维护以外的其他领域，如系统工程和人力资源管理等，更加全面地考虑了组织的整体能力和绩效。但无论是CMM还是CMMI，它们都强调持续过程改进的重要性，旨在帮助组织提高软件的开发和维护过程。



## 七、实施过程改进的IDEAL模型

- 美国卡内基梅隆大学软件工程研究所 (SEI) 的软件过程改进模型IDEAL是软件过程改进框架的一种具体实现。是一个单位用于启动、规划和实现过程改进措施蓝图的模型，概括了建立一个成功的过程改进项目的必要步骤。其中I代表Initiating(启动)、D代表Diagnosing（诊断）、E代表Establishing（建造）、A代表Acting（措施）、L代表Learning（学习）。
- IDEAL模型包括以下五个阶段：



北京大学