



Southeast University

# 软件工程导论

廖力

lliao@seu.edu.cn

# 课程结构

---

**Unit1.** 软件工程概述

**Unit2.** 软件工程技术

**Unit3.** 软件项目管理

一、软件项目管理概述

二、软件度量

三、软件项目估算

四、进度管理

五、风险管理

六、质量管理

七、配置管理

# 一、软件项目管理概述

---

- ❖ 1. 项目管理是通过项目经理和项目组织的努力，运用系统理论的方法对项目及其资源进行计划、组织、协调、控制，旨在实现项目的特定目标的管理方法体系
- ❖ 2. (软件)项目管理的基本内容
  - 项目定义、项目计划、项目执行、项目控制、项目结束

# 一、软件项目管理概述

---

## ❖ 3. 软件项目管理的关注点：4P

### ■ 人员（People）

- 软件开发是人的智力密集的劳动
- 人员是软件工程项目的基本要素和关键因素
- 项目管理必须把人员组织起来以有效地完成软件工作

# 一、软件项目管理概述

---

## ❖ 3. 软件项目管理的关注点：4P

### ■ 产品 (Product)

- 制定项目计划之前，首先确定产品的范围
- 范围包括：目标和范围、可选的解决方案、技术或管理的约束等
- 其目标和范围必须获得软件开发者和客户的一致认可

# 一、软件项目管理概述

---

## ❖ 3. 软件项目管理的关注点：4P

### ■ 过程（Process）

- 在软件过程框架下制定软件开发的综合计划
- 根据项目的特征和项目团队的需求选择框架活动

### ■ 项目（Project）

- 采用科学的方法及工具对项目基本内容进行管理
- 实施有计划，可控制的软件项目

# 一、软件项目管理概述

---

## ❖ 4. To Get to the Essence of a Project: **W<sup>5</sup>HH**

- **Why** is the system being developed?
- **What** will be done?
- **When** will it be accomplished?
- **Who** is responsible?
- **Where** are they organizationally located?
- **How** will the job be done technically and managerially?
- **How much** of each resource (e.g., people, software, tools, database) will be needed?

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

- 软件项目启动
- 建立项目组织
- 项目计划
  - 估算
  - 风险管理
  - 进度安排
- 跟踪与控制
- 软件配置管理
- 软件度量



# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

### ■ 1) 软件项目启动

- 在软件项目启动前对项目进行可行性分析，以明确项目的目标和范围
- 在多个项目方案中选择一个相对完善的方案
  - 考虑交付期限、预算、个人能力、技术界面等限制条件
- 在正式启动软件项目前组成项目组，并召开项目启动会议
  - 内容包括：项目组的初步交流；进一步对项目目标理解；对组织形式、管理方式、方针的一致认识；明确岗位职责

# What is Scope?

---

- ❖ **Scope** refers to all the work involved in creating the products of the project and the processes used to create them. **It defines what is or is not to be done**
- ❖ *Software scope* describes
  - the **functions and features** that are to be delivered to end-users
  - the **data** that are input and output
  - the “content” that is presented to users as a **consequence** of using the software
  - the performance, constraints, interfaces, and reliability that **bound** the system.

# What is Scope?

---

- ❖ **Scope is defined using one of two techniques:**

- A narrative description of software scope is developed after communication with all stakeholders.
- A set of use-cases is developed by end-users.

- ❖ **The project team and stakeholders must have the same understanding of**

- what products will be produced as a result of a project
- how they'll be produced

**This needs good communication!**

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

- 2) 建立项目组织
- 在项目经理领导下，组织不同类型的项目组成员共同协作完成软件项目
- 存在多种可选的项目组织结构，组织结构的选择对项目的成败具有很大影响
  - 封闭式：传统权利层次
  - 随机式：松散团队，适于主动性高人员，进行探索创新型项目
  - 开放式：综合上述两种范型
  - 同步式：项目很自然能划分成若干问题的解决，可同步执行

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

### ■ 2) 建立项目组织

#### ■ 规划软件工程项目组织结构时考虑如下因素：

- 待解决问题的困难程度
- 目标系统的规模，可用代码行或功能点来度量
- 项目组的生存期，即项目小组需要共同工作的时间
- 问题可被分解的程度
- 对目标系统要求的质量和可靠性
- 可供开发时间的紧迫性，即交付时间的严格程度
- 项目组内部的通信的复杂性，即成员(小组)之间正式或非正式通信的机制

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

### ■ 3) 项目计划

**The overall goal of project planning is to **establish a pragmatic strategy** for controlling, tracking, and monitoring a complex technical project.**

**Why?**

***So the end result gets done on time, with quality!***

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

### ■ 3) 项目计划

- 项目计划是项目组织根据软件项目的目标及范围，对项目实施中进行的各项活动进行周密的计划
- 项目计划根据项目目标确定项目的各项任务、安排任务进度、编制完成任务所需的资源预算等
- 项目计划包括：工作计划、人员组织计划、设备采购计划、变更控制计划、进度控制计划、财务计划、文件控制计划、应急计划等

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

### ■ 3.1) 项目估算

#### ■ 项目估算是制定项目计划的基础

- 项目所需的人力(以人月为单位)、项目持续时间(以年份或月份为单位)、成本(以元为单位)等

#### ■ 参照以前类似项目中的相关数据进行估算

- 若存在类似历史项目则可进行类比估算
- 若缺少可类比的项目数据则采用特定的估算技术(例如功能点估算方法等)

#### ■ 通常采用多种估算技术进行交叉检查



# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

### ■ 3.2) 风险管理

- 风险：人员、经费、进度及需求等方面存在的可能影响项目按计划完成的不确定因素
- 风险管理：标识软件项目中的风险，预测风险发生的概率以及风险造成的影响，并对风险进行评估，找出那些可能导致项目失败的风险，然后采取相应的措施来缓解风险
- 风险管理贯彻于整个软件工程过程中

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

### ■ 3.3) 进度安排

- 将项目划分成可管理的子项目、任务和活动
- 确定任务之间的依赖关系，找出影响项目按期完成的关键任务
- 为每个任务分配时间、工作量以及指定责任人，定义每个任务的输出结果及其关联的里程碑
- 在项目实施过程中将在进度计划基础上跟踪实际执行情况，从而及时发现偏差并采取措施加以调整以确保项目按期完成

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

### ■ 4) 跟踪与控制

- 跟踪是控制的前提，它实际上是在项目实施过程中对影响项目进展的内外部因素进行及时的、连续的、系统的记录和报告的活动，其核心在于反映项目变化、提供相关信息的报告
- 控制是通过工具和技术对项目计划与实际执行进行对比，并对项目的未来走向进行预测，再此基础上进行项目的各种调整

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

### ■ 5) 软件配置管理

- **Software Configuration Management (SCM)**  
任务：标识和确定系统中的配置项，在整个生命期内控制这些项的发布和变更，记录并报告配置的状态和变更要求，验证配置项的完整性和正确性
- **SCM**存在于整个软件过程中，是一种保护性活动

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

### ■ 6) 软件度量

- 软件度量是指计算机软件范围内的测量，主要是为产品开发的软件过程和产品本身定义相关的测量方法和标度
  - 对软件开发过程度量的目的是为了对过程进行改进
  - 对产品进行度量的目的是为了产品的质量

# 一、软件项目管理概述

---

## ❖ 5. 软件项目管理过程

- 6) 软件度量
- 度量的作用是为了有效地采用定量的方式来进行管理
- 管理人员利用度量来了解软件工程过程的执行情况 and 产品质量
- 需要考虑：
  - 合适的度量是什么
  - 所收集的数据如何使用
  - 用于比较个人、过程或产品的度量是否合理

## 二、软件度量

---

### ❖ 1. 术语定义

- **Measure(noun 测度)**: 为产品或过程的某些属性的程度、数量、维数、容量或大小提供量化的指示。
  - 收集一个数据点，就建立了一个测度
  - 如：在一个构件的复审中发现的错误数
- **Measure(verb 测量)**: 确定测度的动作(同 measurement)
- **Measurement(测量)**: 确定测度的动作
  - 如：去进行构件评审以收集单元测试错误数
- **Metric(度量)**: 一个系统、构件或过程具有给定属性的量化测量程度
  - 如：每次评审发现错误的平均数

## 二、软件度量

---

### ❖ 1. 术语定义

❖ **Indicator（指标）**：是一个度量或度量的组合，它对软件过程、软件项目或产品本身提供了更深入的了解。

- 如：项目中四个平行小组用不同方法进行了评审，对此评审过程进行度量的结果为：每人/小时所发现的错误数。
- 通过对度量结果的检查，发现采用更加正式的复审方法的两个小组，每人/小时所发现的错误数比起另外两个小组高40%。
- 假设所有其他参数都相同，这就给项目管理者提供了一个指标：正式的复审方法比起其他复审方法在时间投资上能得到更大的回报。



## 二、软件度量

---

### ❖ 2. 度量对象

#### ■ 产品度量

- 如软件产品的结构、模块化程度、复杂性、程序长度等

#### ■ 过程度量

- 测量过程的特定属性，如质量、生产率等，进而间接得到对软件过程的功效的度量，可用于软件过程的改进

#### ■ 项目度量

- 对工作量、时间、错误数量、工作产品交付等进行度量，用以监控项目的进展

## 二、软件度量

---

### ❖ 3. 产品度量的度量领域

- 分析模型的度量
  - 面向功能的度量，面向规模的度量，等
- 体系结构的度量
  - 结构复杂度的度量，面向类的度量，等
- 构件级度量
  - 耦合度度量，内聚度度量，等
- 源代码的度量
  - 复杂性度量，等
- 测试的度量
  - 测试覆盖率度量，与缺陷相关的度量，等

## 二、软件度量

---

### ❖ 4. 面向规模的度量（**Size Oriented Metrics**）

- 软件规模通常是指软件的大小(**size**)，一般用代码行度量
  - 优点：方便、直观
  - 缺点：很大程度上取决于程序设计语言以及软件设计的质量
- 测量出软件规模后可方便地度量其它软件属性，包括：

## 二、软件度量

### ❖4. 面向规模的度量

度量名	含义及表示
LOC或KLOC	代码行数或千行代码数
生产率P	$P=LOC/E$ , E为开发的工作量(常用人月数表示)
每行代码平均成本C	$C=S/LOC$ , S为总成本
文档代码比D	$D=Pe/KLOC$ , 其中Pe为文档页数
代码错误率EQR	$EQR=N/KLOC$ , 其中N为代码中错误数

## 二、软件度量

---


### ❖ 5. 面向功能的度量

- 一种针对软件的功能特性进行度量的方法
- 主要考虑软件系统的“功能性”和“实用性”
- 功能点度量：基于软件信息域的特征(可直接测量)和软件复杂性进行规模度量
- 功能点度量方法步骤：
  - 计算信息域特征的值CT
  - 计算复杂度调整值
  - 计算功能点FP

## 二、软件度量

### ❖ 5. 面向功能的度量

#### ■ 计算信息域特征的值CT

<u>measurement parameter</u>	<u>count</u>	<u>weighting factor</u>				
		<u>simple</u>	<u>avg.</u>	<u>complex</u>		
number of user inputs	<input type="text"/>	X 3	4	6	=	<input type="text"/>
number of user outputs	<input type="text"/>	X 4	5	7	=	<input type="text"/>
number of user inquiries	<input type="text"/>	X 3	4	6	=	<input type="text"/>
number of files	<input type="text"/>	X 7	10	15	=	<input type="text"/>
number of ext.interfaces	<input type="text"/>	X 5	7	10	=	<input type="text"/>
count-total						<input type="text"/>

# 复杂度调整值Fi

	问 题	Fi (0-5)
1	系统需要可靠的备份和恢复吗	
2	需要数据通信吗	
3	有分布处理功能吗	
4	性能很关键吗	
5	系统是否在一个现存的、重负的操作环境中运行	
6	系统需要联机数据登录	
7	联机数据登录是是否需要在多屏幕或多操作之间切换以完成输入	
8	需要联机更新文件吗	
9	输入、输出、文件或查询很复杂吗	
10	内部处理复杂吗	
11	代码需要被设计成可复用的吗	
12	设计中需要包括转换及安装吗	
13	系统的设计支持不同组织的多次安装吗	
14	应用的设计方便用户修改和使用吗	
总 计		

## 二、软件度量

### ❖ 5. 面向功能的度量

### ❖ 功能点计算公式 $FP = CT * (0.65 + 0.01 * F)$

- 其中：CT前面的信息域特征值之和，F是前面Fi之和

### ❖ 功能点可用于：

度量名	含义表示
生产率P	$P = FP / E$ ，E为开发的工作量（常用人月数表示）
每个功能点成本C	$C = S / FP$ ，S为总成本
每个功能点文档数D	$D = P_e / FP$ ，其中Pe为文档页数
功能点错误率EQR	$EQR = N / FP$ ，其中N为错误数



## 二、软件度量

### ❖ 6. 功能点与LOC的换算(部分)

程序语言	每FP之LOC值			
	平均	中等	低	高
Access	35	38	15	47
Ada	154	—	104	205
APS	86	83	20	184
ASP	62	—	32	127
Assembler	337	315	91	694
C	162	109	33	704
C++	66	53	29	178
Java	63	53	77	70
COBOL	77	77	14	400
SQL	40	37	7	110
VBScript	34	27	50	—
Visual Basic	47	42	16	158

# 三、软件项目估算

---

- ❖ 1. 软件项目估算的任务：估算成本和工作量
  - 分解问题（**WBS**）
  - 使用两种以上不同的估算方法进行估算
  - 调和不同估算

## 三、软件项目估算——2.WBS

---

### ❖ 1) 定义:

- **A work breakdown structure (WBS) is a deliverable-oriented grouping of the work involved in a project that defines the total scope of the project**

### ❖ 2) WBS书写格式:

- 树形结构
- 目录结构

## 三、软件项目估算——2.WBS

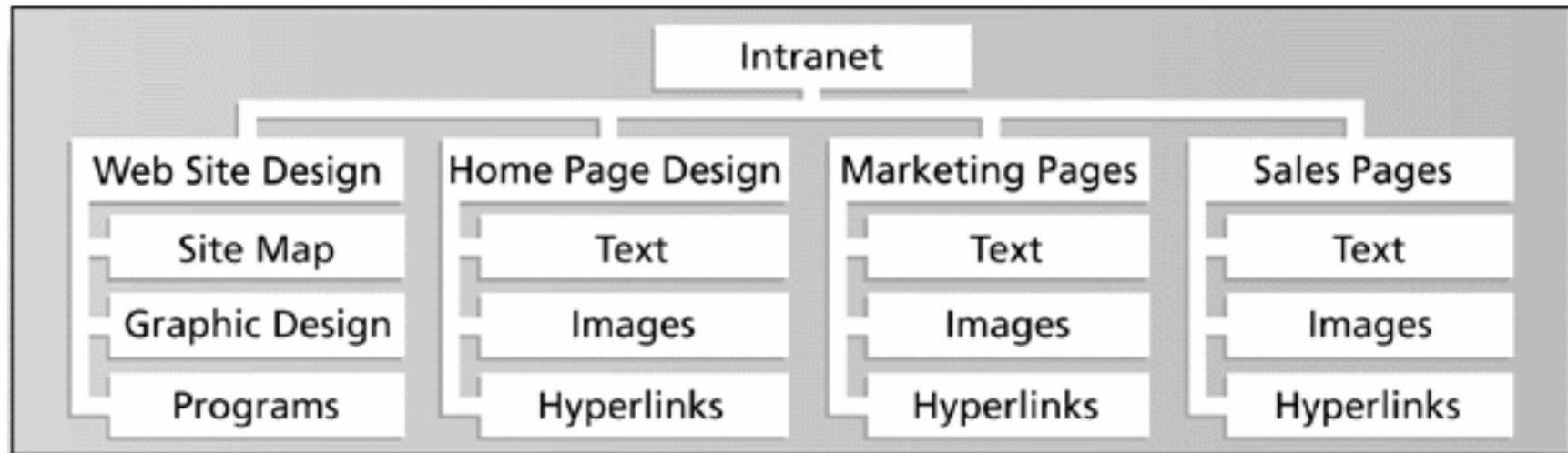
---

### ❖ 3) WBS的分解:

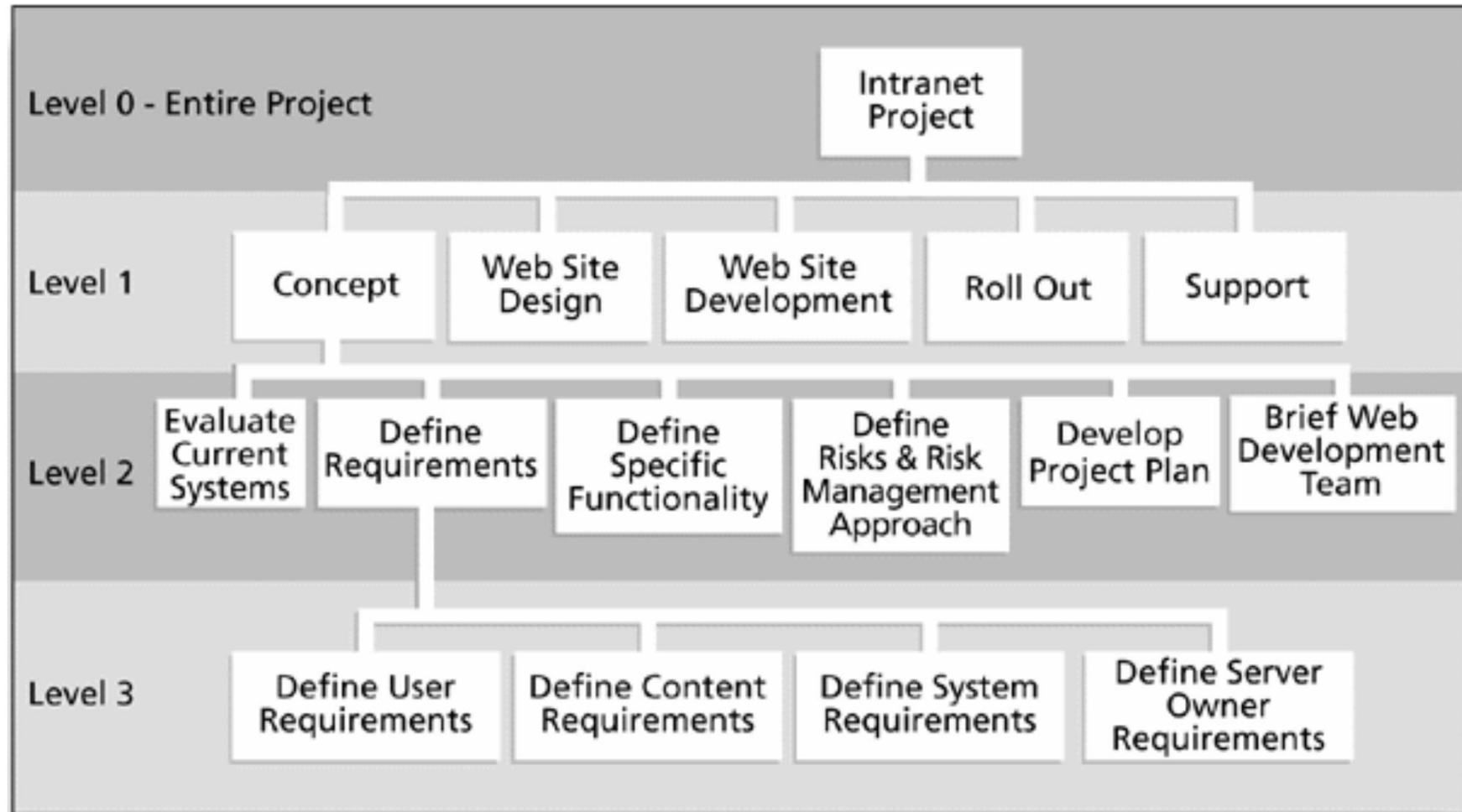
- **WBS is usually organized around**
  - **project products (tools, tangible outputs, .)**
  - **project phases (concepts, steps,..)**
  - **project management process groups (initializing, developing, executing, closing)**
- **因此，其分解方式为**
  - **产品分解：** 将一个复杂问题分解成若干个小问题
  - **过程分解：** 根据软件开发过程中的活动(分析、设计、编码、测试等)进行分解

# Example: 树形结构的产品分解

---



# Example: 树形结构的过程分解



# **Example: 目录结构的过程分解**

---

## **1.0 Concept**

**1.1 Evaluate current systems**

**1.2 Define Requirements**

**1.2.1 Define user requirements**

**1.2.2 Define content requirements**

**1.2.3 Define system requirements**

**1.2.4 Define server owner requirements**

**1.3 Define specific functionality**

**1.4 Define risks and risk management approach**

**1.5 Develop project plan**

**1.6 Brief Web development team**

**2.0 Web Site Design**

**3.0 Web Site Development**

**4.0 Roll Out**

**5.0 Support**

## 三、软件项目估算——2.WBS

---

### ❖ 4) 工作包 (Work Packages)

- **WBS结构的叶子节点，即任务分解的最小单位，不可再划分的子任务。**
- **划分原则：**
  - **Use the “one-to-two” rule: 1 or 2 persons for 1 or 2 weeks**
  - **Ideally shorter rather than longer: (2-3 weeks maximum, 1 day minimum)**
  - **Basis for monitoring and reporting progress**
    - **Can be tied to budget items (with charge numbers)**
    - **Resources (personnel) assigned**



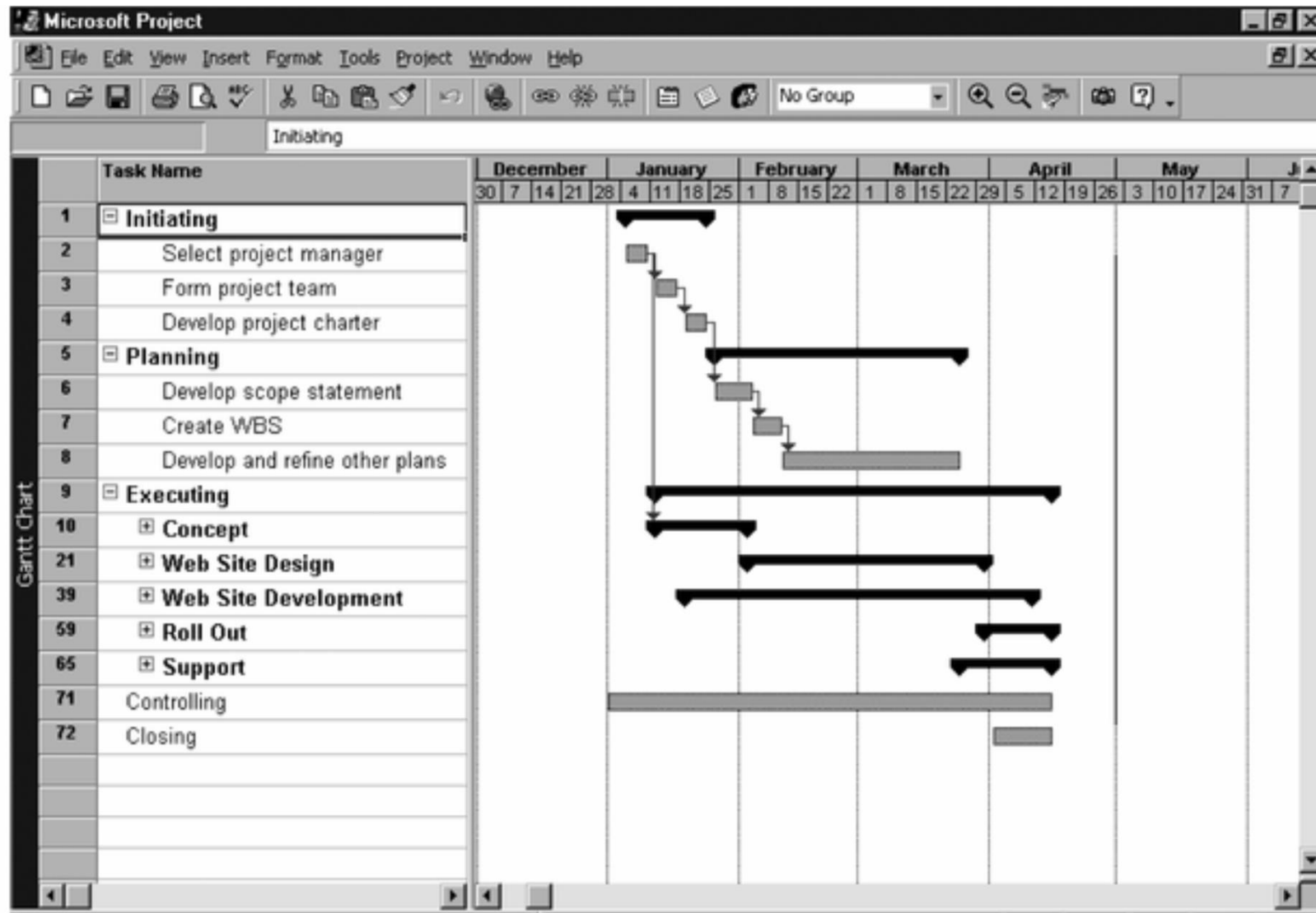
## 三、软件项目估算——2.WBS

---

### ❖ 5) WBS的作用

- It provides the **basis** for
  - Planning and managing project schedules, costs, and changes
  - Risk analysis
  - Organizational structure
  - Measurement

# Example WBS Organized by Project Management Process Groups



## 三、软件项目估算——2.WBS

---



**If it's not in the WBS,  
It's not in the project !**

# 三、软件项目估算

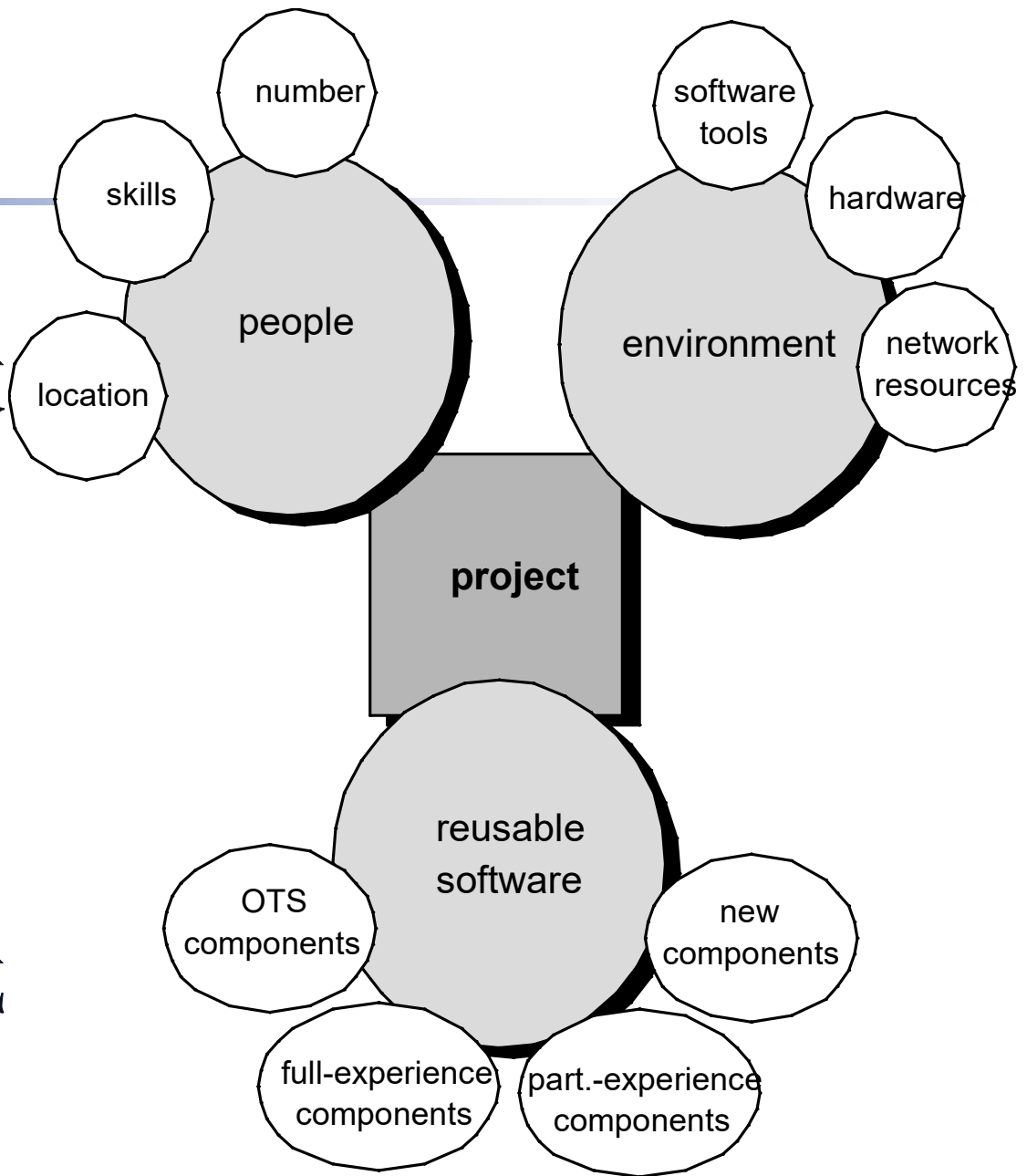
## ❖ 3. 估算概述

### ■ 1) 对软件开发所需的资源进行估算

- 资源描述
- 可用性说明
- 何时需要
- 持续时间

### ■ 2) 对项目的成本进行估算

- 人月



## 三、软件项目估算

---

### ❖ 4.成本估算的常用方法:

- 1) 基于已经完成的类似项目进行估算, 这是一种常用的也是有效的估算方法
- 2) 基于规模的估算
  - 基于LOC的估算
  - 基于FP的估算
- 3) 基于经验估算模型的估算
  - **COCOMO**模型, 等

## 三、软件项目估算——4. 估算方法

---

### ❖ 2) 基于规模的估算

- **LOC估算和FP估算是两种不同估算技术，不过具有以下共性：**
  - 均以问题分解为基础。**LOC**尤甚，问题分解得越详细越好。
  - 要求**WBS**中每个工作包都是可以分别独立估算的
  - 每个工作包都有**基线生产率**度量值可以参考，或者可以预估其生产率。
  - **基线生产率度量：**根据对**LOC**和**FP**的测量，计算出软件开发组织对某类问题的生产率，**LOC/pm** 或者**FP/pm**。

## 三、软件项目估算——4. 估算方法

---

### ❖ 2) 基于规模的估算

#### ■ 估算方法

- **Step1.** 项目计划人员为每个功能或信息域确定一个估算值的范围：乐观值**opt**，可能值**m**，悲观值**pess**
- **Step2.** 计算各个功能或信息域的规模的期望值  
$$-S=(S_{opt} + 4S_m + S_{pess})/6$$
- **Step3.** 应用历史生产率数据估算项目的成本

## 三、软件项目估算——4. 估算方法

---

### ❖ 3) 基于经验估算模型的估算

- 基于经验模型的估算针对性强，精度高，应用广泛
- 如：**COCOMO (Constructive Cost Modeling)**
  - 不仅可应用于软件工程早期阶段预估工作量和成本
  - 也可在设计构造阶段应用，以对项目的进度和人员安排作出调整



## 四、项目进度管理

---

- ❖ 1. 项目进度管理的目标
- ❖ 确保软件项目在规定的时间内按期完成
  - 进度延误是项目失败的最主要原因之一
  - 项目进度延误的原因：
    - **an unrealistic deadline**
    - **changing customer requirements**
    - **an honest underestimate (低估) of the amount of effort and/or the number of resources**

# 四、项目进度管理

---

## ❖ 1. 项目进度管理的目标

### ■ 项目进度延误的原因（续）：

- **predictable and/or unpredictable risks**
- **technical difficulties**
- **human difficulties**
- **miscommunication among project staff**
- **a failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem**

# 四、项目进度管理

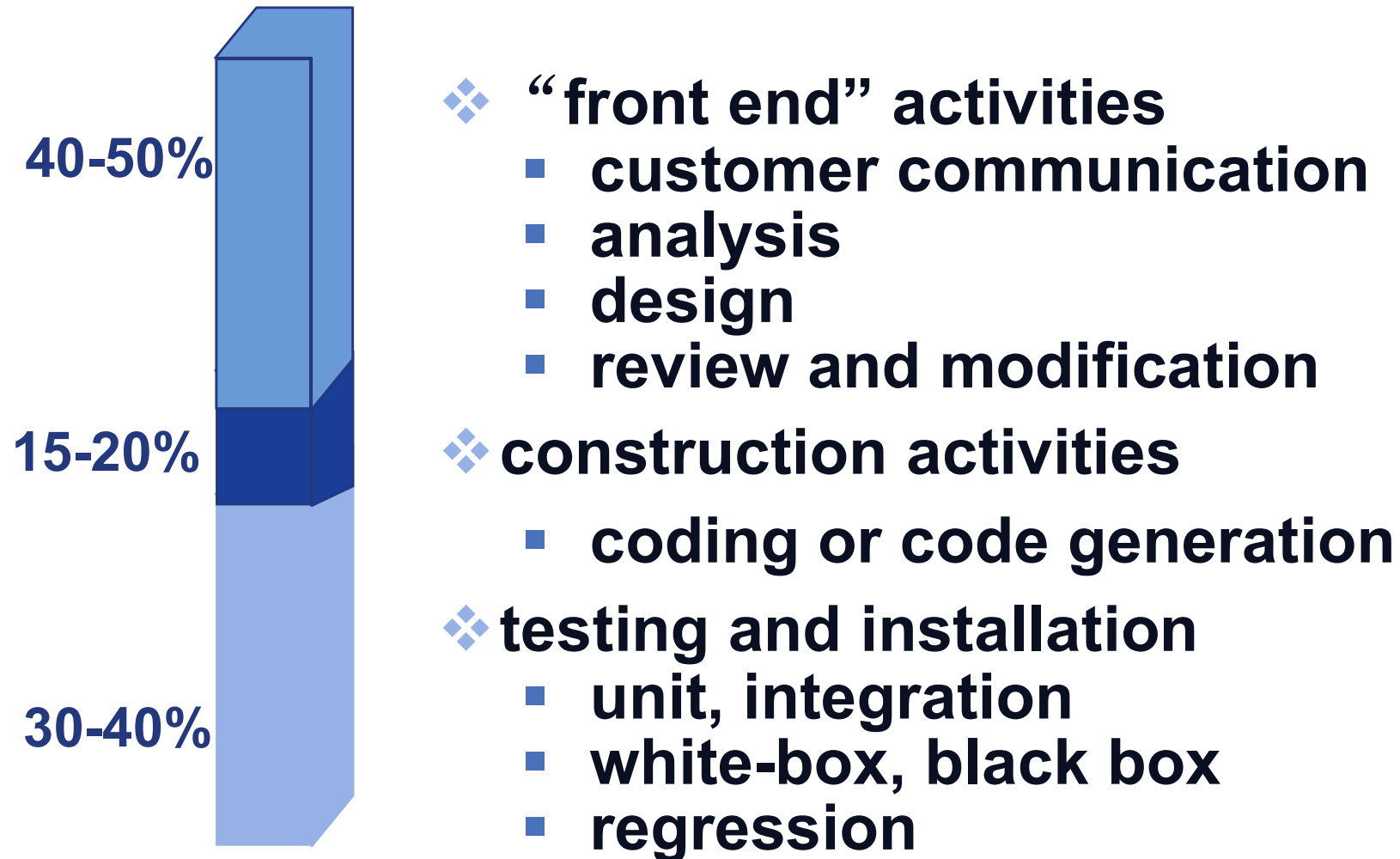
---

## ❖ 2. Scheduling Principles

- **compartmentalization**—define distinct tasks
- **interdependency**—indicate task interrelationship
- **effort validation** —be sure resources are available
- **defined responsibilities** —people must be assigned
- **defined outcomes** —each task must have an output
- **defined milestones** —review for quality

# Effort Allocation

---



# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

- **1) Defining Task Sets——WBS**
  - 定义所有的项目任务
- **2) Sequencing Activities**
  - 确定项目任务之间的依赖关系
- **3) Drawing Project network diagrams**
  - 规划每个任务所需的工作量和持续时间，构建项目任务网络
- **4) Critical path analysis （关键路径分析）**
  - 找出影响项目进度的关键活动

# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

- **5) Using Gantt charts (甘特图) for scheduling**
  - 定义人员责任，任务结果，确定里程碑
  - 通过适当工具建立进度计划
- **6) Schedule Tracking**
  - 在项目开发过程中不断跟踪项目的执行情况，发现那些未按计划进度完成的任务对整个项目工期的影响，并及时进行调整

# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

- 1) Defining Task Sets
- 项目进度计划要制定出的一个周密的项目活动安排。要求：
  - 根据项目的目标
  - 在项目确定的范围内
  - 依据确定的需求和质量标准
  - 在项目成本预算许可下
- 因此，要制定项目进度计划首先要进行任务划分。

# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

### ■ 1) Defining Task Sets

- 项目必须被划分成若干可以管理的活动和任务：
  - 根据不同的软件项目性质，选择合适的软件工程过程
  - 对产品和过程都进行分解
  - 项目分解的工具是工作分解结构（**Work Breakdown Structure**），它是一个分级的、面向可交付成果的树型结构



## 1.1 Concept scoping determines the overall scope of the project.

Task definition: Task 1.1 Concept Scoping

1.1.1 Identify need, benefits and potential customers;

1.1.2 Define desired output/control and input events that drive the application;

Begin Task 1.1.2

1.1.2.1 FTR: Review written description of need

FTR indicates that a formal technical review (Chapter 26) is to be conducted.

1.1.2.2 Derive a list of customer visible outputs/inputs

1.1.2.3 FTR: Review outputs/inputs with customer and revise as required;

endtask Task 1.1.2

1.1.3 Define the functionality/behavior for each major function;

Begin Task 1.1.3

1.1.3.1 FTR: Review output and input data objects derived in task 1.1.2;

1.1.3.2 Derive a model of functions/behaviors;

1.1.3.3 FTR: Review functions/behaviors with customer and revise as required;

endtask Task 1.1.3

1.1.4 Isolate those elements of the technology to be implemented in software;

1.1.5 Research availability of existing software;

1.1.6 Define technical feasibility;

1.1.7 Make quick estimate of size;

1.1.8 Create a Scope Definition;

endTask definition: Task 1.1

is  
refined  
to





# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

### ■ 2) Sequencing Activities

- 确定各个被划分的活动或任务之间的相互关系
  - 有些任务必须是串行的
  - 有些可能是并行的

DEPENDENCY TYPE	ABBREVIATION	EXPLANATION	HOW IT LOOKS ON A GANTT CHART	EXAMPLE
Finish-to-Start	FS	Task 1 must finish before task 2 can start.		A computer must be installed (task 1) before application software can be installed (task 2).
Start-to-Start	SS	Task 1 must start before task 2 can start.		As soon as you start installing hardware (task 1), you can start documenting serial numbers (task 2).
Finish-to-Finish	FF	Task 1 must finish before task 2 can finish.		A computer backup (task 1) must be finished before the shutdown of the system is completed (task 2).
Start-to-Finish	SF	Task 1 must start before task 2 can finish.		In the event of a power interruption, the UPS must start (task 1) before the operator can finish shutting down the system in an orderly fashion (task 2).

# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

### ■ 3) Drawing Project network diagrams

#### ■ 时间分配

- 必须为每个被调度的任务分配一定数量的工作单位（如人天）
- 还必须为每个任务制定开始和结束日期，这些日期是相互依赖的

#### ■ 工作量确认

- 根据软件工程过程的不同，可确定其相应的任务的工程量分配。
- 确保在任意时段中分配给任务的人员数量不会超过项目组中的人员数量

# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

- 3) Drawing Project network diagrams
- 工作量确认中需注意：参与项目的人员数与整体生产率之间的关系并非是线性的
  - 存在人员之间的交流开销：一个由 $n$ 个人组成的项目组内共存在 $n(n-1)/2$ 条通信路径
  - 增加一个人并不等于净增了一个人的工作量，应扣除相应的通信代价
  - 每个开发小组的成员不宜太多，通过合理的组织形式减少组内的通信路径数

# 四、项目进度管理

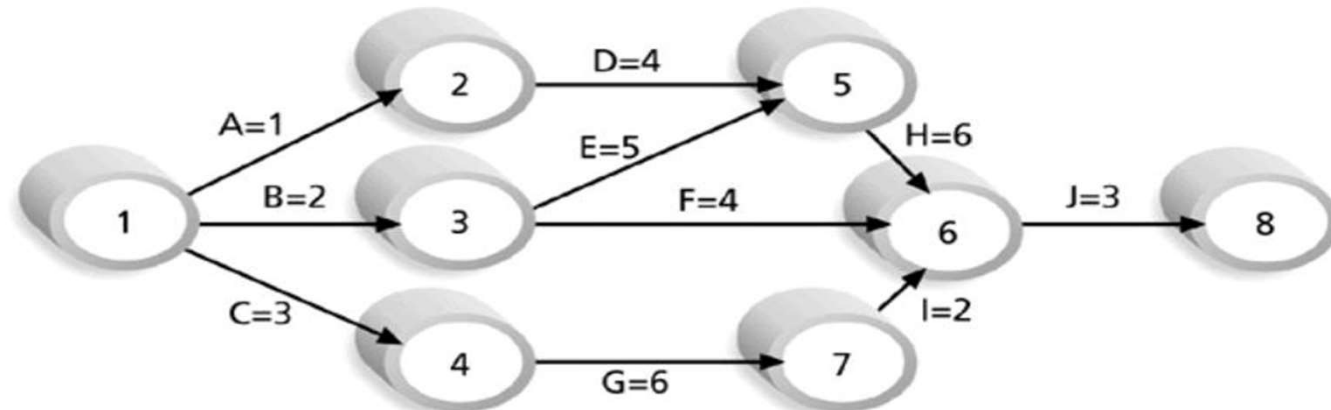
---

## ❖ 3. Scheduling Steps

- 3) Drawing Project network diagrams
- 构建项目任务网络（Project network diagrams）
  - Project network diagrams are good at for showing activity sequencing and interdependencies.
  - Arrow Diagramming Method (ADM)
  - Precedence Diagramming Method (PDM)

# Arrow Diagramming Method (ADM)

- ❖ Also called **activity-on-arrow (AOA)** network diagrams
  - Activities are represented by arrows
  - Nodes or circles are the starting and ending points of activities
  - Can only show **finish-to-start** dependencies



Note: Assume all durations are in days; A=1 means Activity A has a duration of 1 day.

# **Precedence Diagramming Method (PDM)**

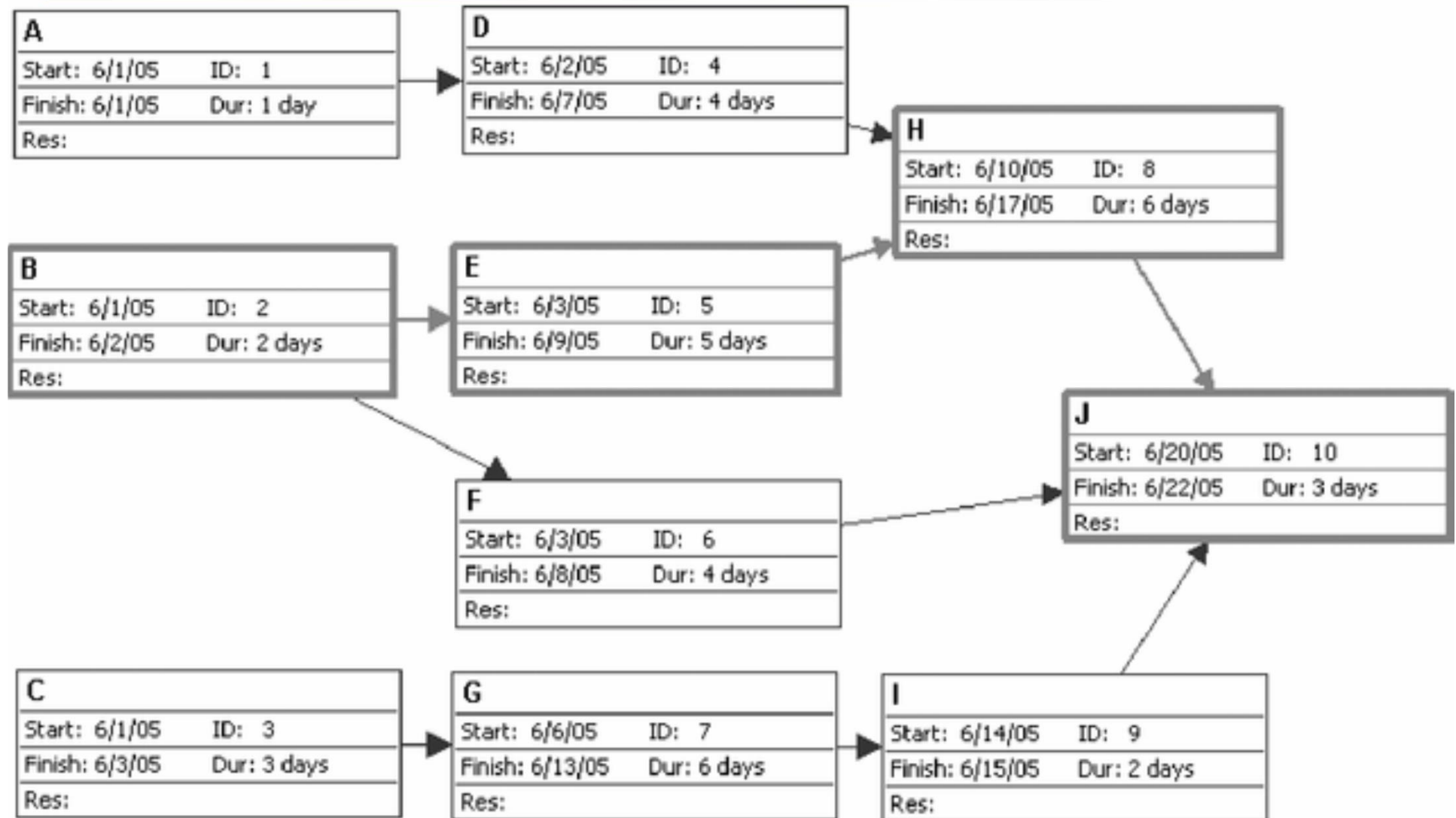
---

❖ **Also called activity-on-node (AON)  
network diagrams**

- **Activities are represented by boxes**
- **Arrows show relationships (dependencies) between activities**
- **More popular than ADM method and used by project management software**
- **Better at showing different types of dependencies**



# PDM



# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

- **3) Drawing Project network diagrams**
- **任务网络的优点**
  - **Show precedence (优先关系) well**
  - **Ability to identify critical path （关键路径分析）**
  - **Ability to perform “what if” analysis**

# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

- **3) Drawing Project network diagrams**
- **任务网络的缺点**
  - **Default model assumes resources are unlimited**
    - **we need to incorporate Resource Dependencies ourselves when determining the “real” Critical Path**
  - **Difficult to follow on large projects**

# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

- **4) Critical path analysis** （关键路径分析）
- **机动时间**：在不影响整个工期的情况下，当前任务允许延迟的最长时间
- 通过计算每个任务的机动时间来求项目的**关键路径**

# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

- 4) **Critical path analysis** （关键路径分析）
- 方法步骤：
  - 计算最早时刻**EFT**
  - 计算最迟时刻**LET**
  - 计算机动时间
  - 得出关键路径：由机动时间为**0**的任务组成的项目路径

# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

- **5) Using Gantt charts (甘特图) for scheduling**
- 为监控软件项目的进度计划和工作实际进展情况，表现各项任务之间进度的相互依赖关系
- 需要采用图示的方法明确标识：
  - 各个任务的计划开始时间和完成时间
  - 各个任务的完成标志
  - 各个任务与参与工作的人数，各个任务与工作量之间的衔接情况
  - 完成各个任务所需的物理资源和数据资源

## 四、项目进度管理

---

### ❖ 3.Scheduling Steps

- **5) Using Gantt charts (甘特图) for scheduling**
- **甘特图(Gantt Chart)也称时间表(Timeline chart)，用来建立项目进度表**
- **在甘特图中，每项任务的完成以必须交付的文档和通过评审为标准，因此它们往往作为项目里程碑**

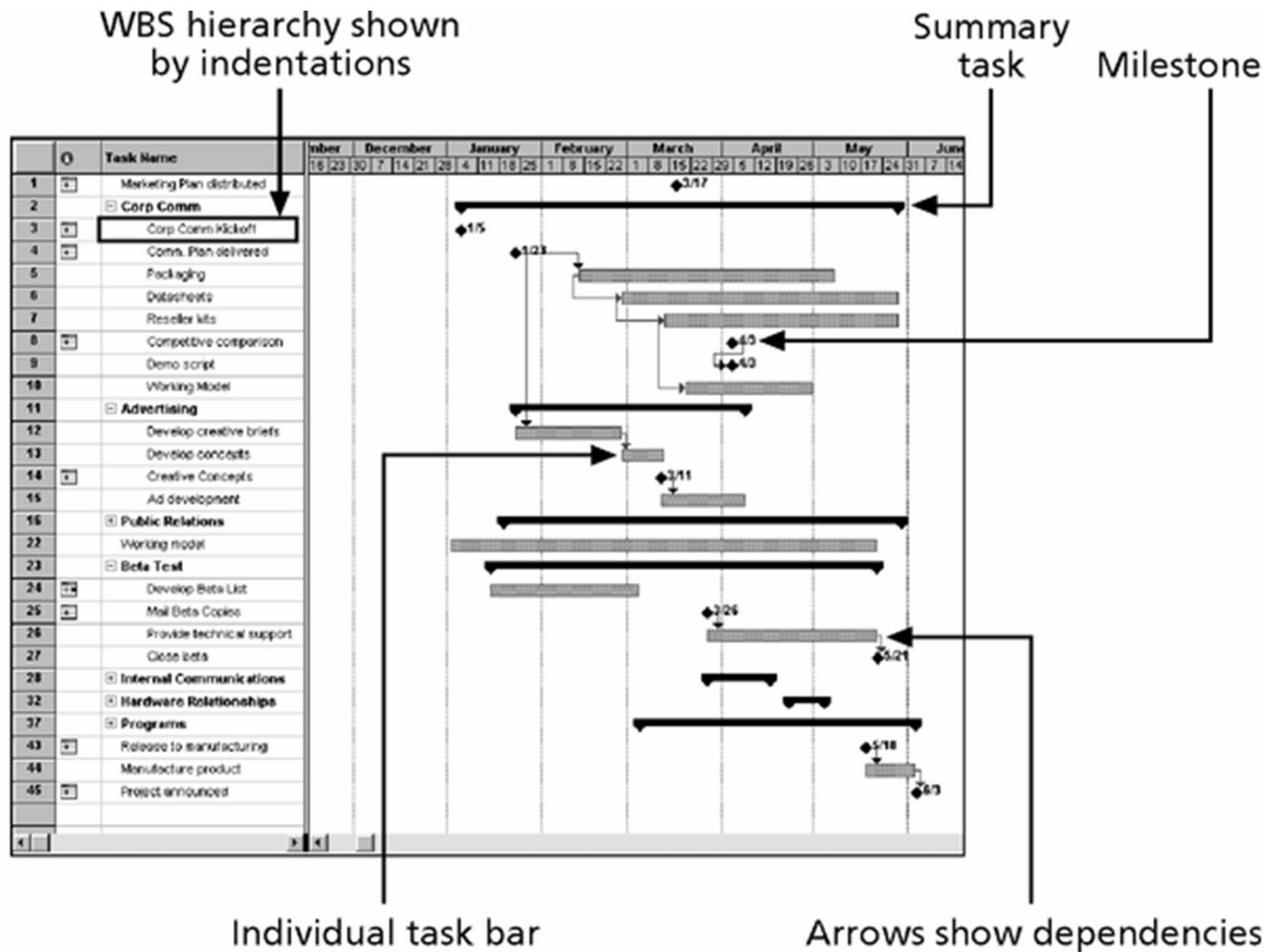
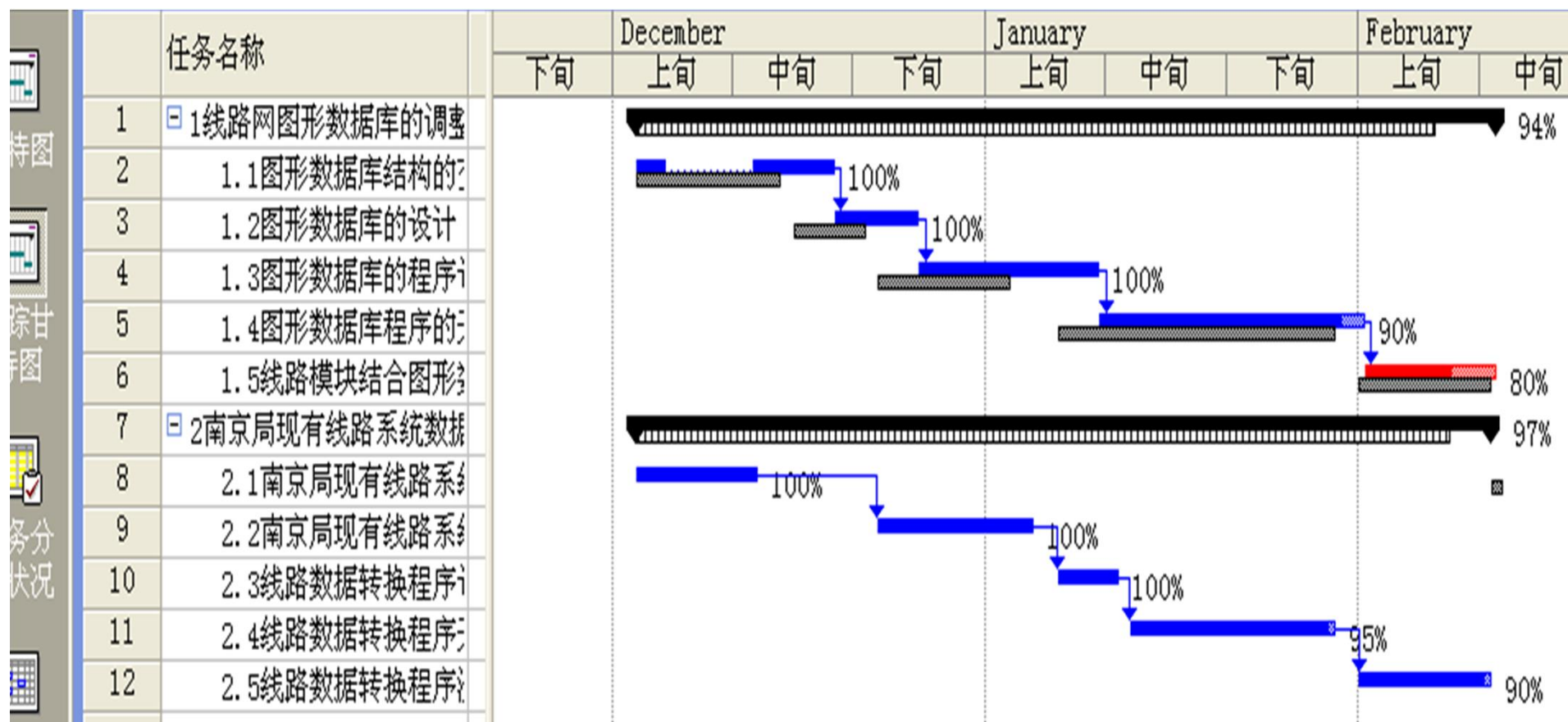


Figure 6-6. Gantt Chart for Software Launch Project



# 甘特图示例



# 项目任务列表

		①	任务名称	工期	开始时间	完成时间
甘特图	1		▣ 1线路网图形数据库的调整 IBM RS6000小型机1台	54.38 工作日	2001年12月3日	2002年2月12日
	2	✓	▣ 1.1图形数据库结构的交流	8 工作日	2001年12月3日	2001年12月19日
跟踪甘特图	3	✓	▣ 1.2图形数据库的设计 研发工程师3人	6 工作日	2001年12月19日	2001年12月26日
	4	✓	▣ 1.3图形数据库的程序设计 研发工程师5人	11 工作日	2001年12月26日	2002年1月10日
任务分配状况	5	▣	▣ 1.4图形数据库程序的开发 研发工程师2人	17 工作日	2002年1月10日	2002年2月1日
	6	▣	▣ 1.5线路模块结合图形数据库进行测试 测试工程师2人	7 工作日	2002年2月1日	2002年2月12日
日历	7		▣ 2南京局现有线路系统数据的转换 IBM RS6000小型机1台	54 工作日	2001年12月3日	2002年2月11日
	8	✓	▣ 2.1南京局现有线路系统数据库结构的交流 南京局线路组1人提供数据库结构的材料 系统工程师2人	8 工作日	2001年12月3日	2001年12月12日
网络图	9	✓	▣ 2.2南京局现有线路系统数据库结构分析 南京局线路组1人配合 系统工程师2人	11 工作日	2001年12月23日	2002年1月4日
	10	✓	▣ 2.3线路数据转换程序设计 软件工程师5人	5 工作日	2002年1月7日	2002年1月11日
资源工作表	11	▣	▣ 2.4线路数据转换程序开发 研发工程师3人	13 工作日	2002年1月13日	2002年1月29日
	12	▣	▣ 2.5线路数据转换程序测试 测试工程师2人	7 工作日	2002年2月1日	2002年2月11日

# 四、项目进度管理

---

## ❖ 3.Scheduling Steps

### ■ 6) Schedule Tracking

- 根据项目进度表，跟踪和控制各任务的实际执行情况
- 一旦发现某个任务(特别是关键路径上的任务)未在计划进度规定的时间范围内完成，那么就要采取措施进行调整
  - 增加额外的资源、增加新的员工或调整项目进度表

## 四、项目进度管理

---

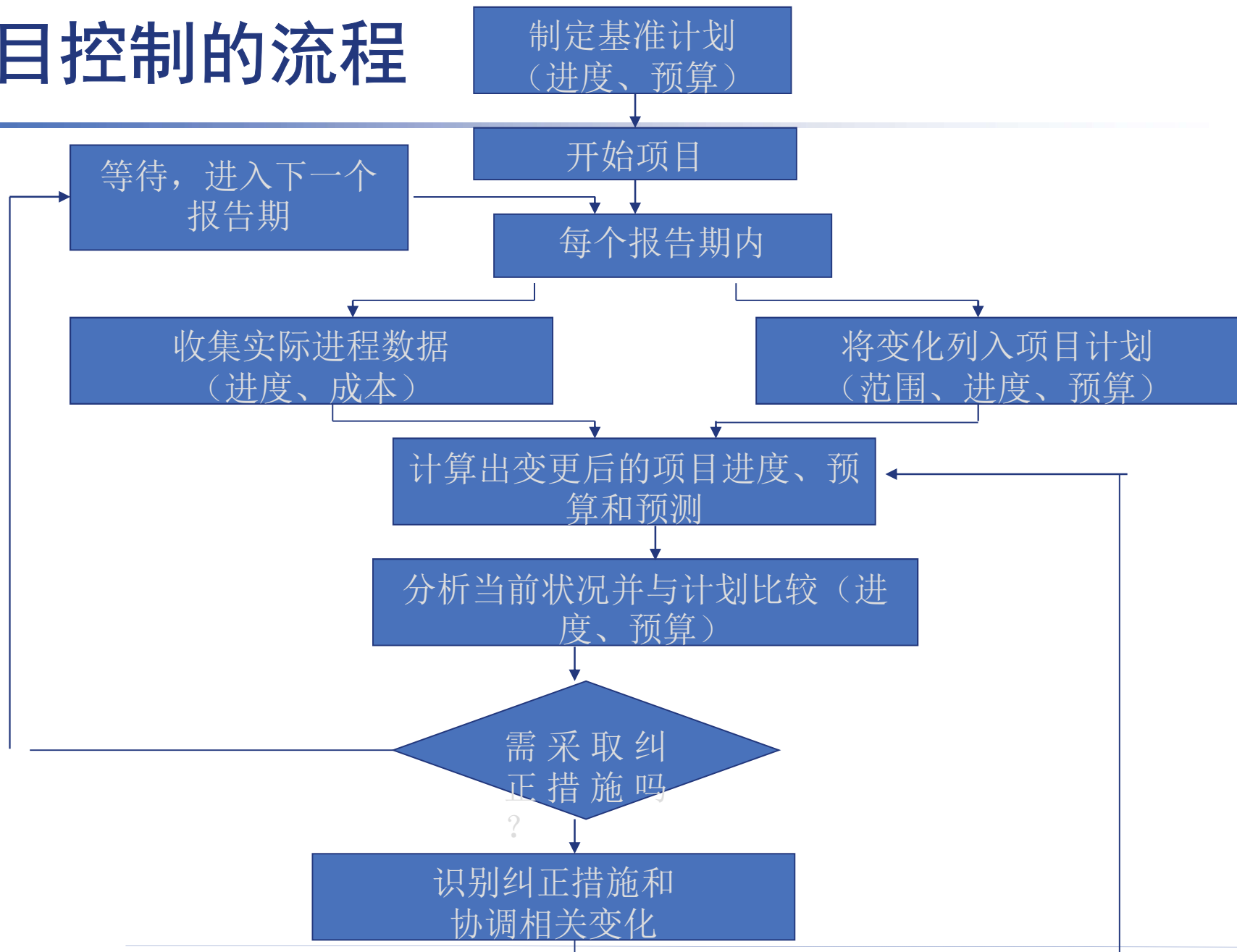
### ❖ 3.Scheduling Steps

#### ■ 6) Schedule Tracking

#### ■ 可以通过以下方式来实现项目跟踪：

- 定期举行项目状态会议，由项目组中的各个成员分别报告进度和问题
- 评价在软件工程过程中产生的所有评审结果
- 确定正式的项目里程碑是否在预定日期内完成
- 比较项目表中列出的各项任务的实际开始日期与计划开始日期
- 非正式与开发人员进行会谈，获取他们对项目进展及可能出现的问题的客观评价

# 项目控制的流程



# 五、风险管理

---

- ❖ 现代项目管理与传统项目的不同之处就是引入了风险管理技术
- ❖ 1. 什么是风险？
  - 在给定情况下和特定时间内，那些可能发生的结果与预期结果之间的差异，差异越大，风险越大



# 五、风险管理

---

## ❖ 2. 风险的类别

### ■ 项目风险

- 可能对项目的预算、进度、人力、资源、顾客和需求等方面产生不良影响的潜在问题

### ■ 技术风险

- 潜在的设计、实现、接口、验证和维护等方面的问题
- 规约的二义性
- 技术的不确定性
- 陈旧或不成熟的“领先的”技术

# 五、风险管理

---

## ❖ 2. 风险的类别

- **商业风险：** 威胁要开发的软件的生存能力
  - 开发了一个无人真正需要的产品(市场风险)
  - 开发的产品不符合公司的整体商业策略(策略风险)
  - 建造了一个销售部门不知如何销售的产品
  - 由于重点转移失去了高级管理层支持(管理风险)
  - 没有得到充分预算或人力资源保证(预算风险)



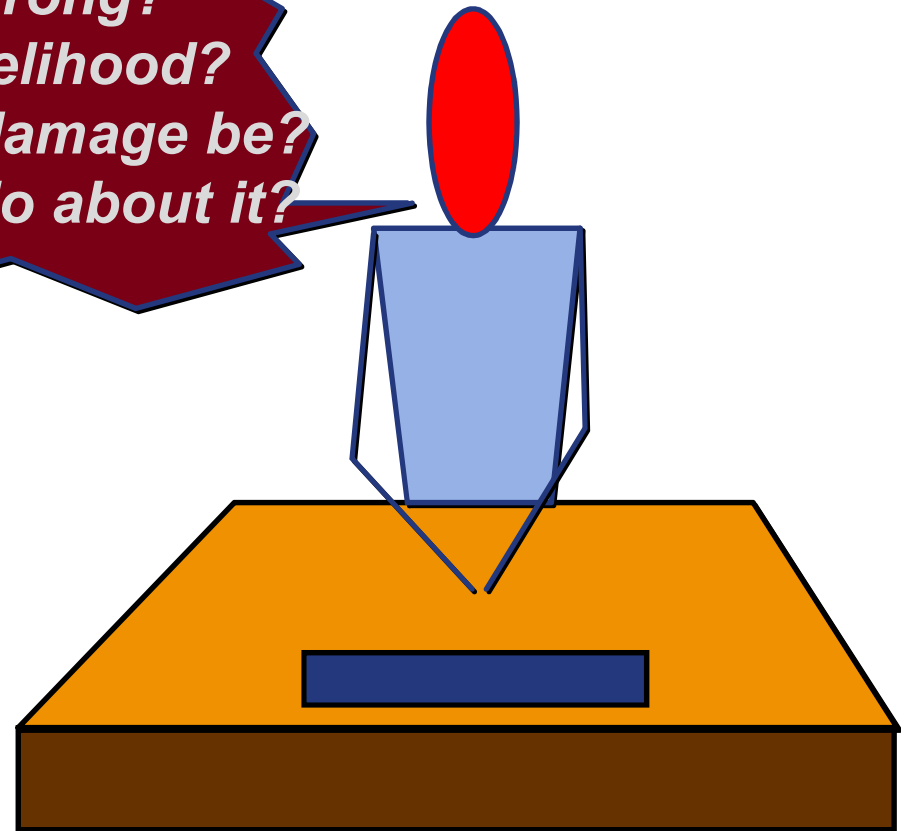
# 五、风险管理

---

*What can go wrong?  
What is the likelihood?  
What will the damage be?  
What can we do about it?*

## ❖ 3. 什么是风险管理？

- 风险管理就是识别评估风险，建立、选择、管理和解决风险的可选方案和组织方法



## 五、风险管理——4. 主动 vs. 被动

---

### ❖ Reactive Risk Management

- project team reacts to risks when they occur
- mitigation—plan for additional resources in anticipation of fire fighting
- fix on failure—resource are found and applied when the risk strikes
- crisis management—failure does not respond to applied resources and project is in jeopardy (危险)

## 五、风险管理——4. 主动 vs. 被动

---

### ❖ Proactive Risk Management

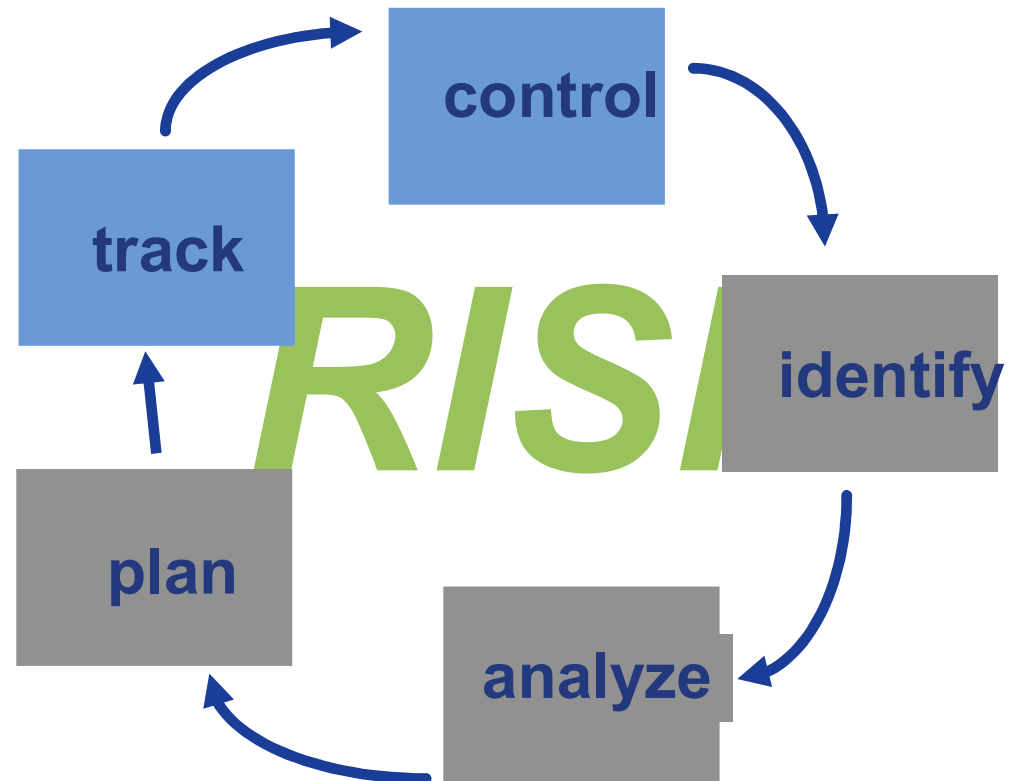
- formal risk analysis is performed
- organization corrects the root causes of risk
  - TQM concepts and statistical SQA
  - examining risk sources that lie beyond the bounds of the software
  - developing the skill to manage change

# 五、风险管理

---

## ❖ 5. 主动风险管理的过程

- 风险识别
- 风险分析
- 风险规划
- 风险监控



## 五、风险管理——5. 主动风险管理

---

### ❖ 1) 风险识别

#### ❖ 主要识别下列类型中的已知的和可预测的风险

- **Product size:** 总体规模
- **Business impact:** 业务约束, 市场约束
- **Customer characteristics:** 客户素质
- **Process definition:** 过程相关
- **Development environment:** 开发工具
- **Technology to be built:** 复杂性, 创新性
- **Staff size and experience:** 人员

# 五、风险管理——5. 主动风险管理

---

## ❖ 1) 风险识别

### ❖ 风险因素

- 性能风险：产品能满足需求且符合其使用目的的不确定的程度
- 成本风险：项目能否被维持的不确定的程度
- 支持风险：软件易于维护的不确定的程度
- 进度风险：项目进度能被维持且产品能按时交付的不确定的程度

# 五、风险管理——5. 主动风险管理

---

## ❖ 2) 风险分析

### ❖ A. 风险预测:

- 评价每种风险发生的可能性或概率以及当该风险发生时所导致的后果
- 风险预测任务:
  - 建立一个尺度, 以反映风险发生的可能性
  - 描述风险的后果
  - 估算风险对项目及产品的影响
  - 标注风险预测的整体精确度以免产生误解

# 五、风险管理——5. 主动风险管理

---

## ❖ 2) 风险分析

### ❖ B. 建立风险表

- 第1列列出所有的风险
- 第2至4列列出每个风险的种类(项目风险、技术风险、商业风险等)、发生的概率以及所产生的影响
- 综合考虑风险发生的概率和风险所产生的影响，对风险表排序



# Building a Risk Table

风险	类型	概率	影响值	RMMM
用户数量可能大大超出计划	产品规模	30%	3	...
...	...	...	...	...

# 五、风险管理——5. 主动风险管理

---

## ❖ 2) 风险分析

## ❖ C. 风险评估

- 进一步审查风险预测阶段对各种风险预测的精确度，并定义一条中截线
  - 中截线：风险表中某处的一条水平线
  - 中截线以上是会进一步关注的高风险
  - 中截线以下的风险则需要重新评估及二次排序。
  - 所有在中截线以上的风险都必须进行管理
  - 风险表的最后一列**RMMM**为其管理进行指示

# 五、风险管理——5. 主动风险管理

---

## ❖ 2) 风险分析

## ❖ C. 风险评估

### ■ 评估风险的影响

- 计算风险显露度 (**Risk Exposure**)
- **RE=P\*C**, **P**: 风险发生的概率
- **C**: 风险发生时带来的项目成本
- 风险显露度为调整项目最终的成本估算提供了依据。
- 还可预测开发各阶段所需人员的增长。

# 五、风险管理——5. 主动风险管理

---

## ❖ 3) 风险规划(RMMM)

- 风险分析的主要目的就是辅助项目团队制定处理风险的策略
- 有效的风险规划必须考虑：
  - 风险避免 (Risk Mitigation)
  - 风险监测 (Risk Monitoring)
  - 风险管理及应急计划 (Risk Management Plan)

# 风险避免 (Risk Mitigation)

---

- ❖ 对付风险的最好办法是主动地避免风险
- ❖ 即：
  - 在风险发生前，分析引起风险的原因，然后采取措施，以避免风险的发生

# 风险避免（Risk Mitigation）

---

- ❖ 例如为避免“频繁的人员流动”风险可采取如下策略：
  - 与现有人员探讨人员流动的原因
  - 采取行动，缓解那些管理控制范围内的原因
  - 一旦项目启动，采取一些技术来保证在人员离开时工作的连续性
  - 对项目组进行良好的组织
  - 定义文档的标准并建立机制确保文档能被及时建立
  - 通过评审会议进行培训
  - 对关键的技术人员都配备后备

# 风险监控（Risk Monitoring）

---

- ❖ 监控可以提供风险指示(是否正在变高或变低)的因素
- ❖ 监控目的：
  - 评估所预测的风险是否真的发生了
  - 保证正确实施了各项缓解步骤
  - 收集能够用于今后风险分析的信息

# 风险监控（Risk Monitoring）

---

❖ 例如，对人员流动风险可监控如下因素：

- 项目组成员对项目的态度
- 项目组的凝聚力
- 成员之间的关系
- 与报酬和利益相关的问题
- 在公司内和公司外工作的可能性



# 风险管理及应急计划 (RMP)

---

## ❖ RMP实施的前提条件

- 风险避免和缓解工作已经失败
- 风险已发生

## ❖ 例如，对人员流动风险的应急措施：

- 后备人员转正
- 重新调整资源和进度，使得新加入团队的人员能赶上进度
- 要求即将离开的人员进入知识交接模式

## 五、风险管理——5. 主动风险管理

---

### ❖ 3) 风险规划

- 对于每个中截线以上的风险，都应制定 **RMMM**
- **RMMM**的实施会导致额外的项目开销
  - 如上例中项目后备人员等都会导致额外开销。

# RMMM目录

---

## 1.引言

### 1.1文档的范围和目的

### 1.2概述

#### 1)目标

#### 2)风险转化的优先级

### 1.3组织

#### 1)管理

#### 2)职责

#### 3)工作流程

### 1.4风险转化过程

#### 1)进度

#### 2)里程碑和评审

#### 3)预算

## 2.风险分析

### 2.1风险识别

#### 1)风险源及风险概述

#### 2)风险分类

## 2.2 风险预测

### 1)估算风险概率

### 2)估算风险的后果

### 3)估算规则

### 4)产生估计误差的原因

## 2.3风险评估

### 1) 评估方法

### 2) 评估假设及限制性

### 3) 评估结果

## 3.风险管理

### 3.1 建议

### 3.2 风险转化选项

### 3.3 控制风险转化的建议

### 3.4 风险监控过程

## 4. 附录

### 4.1 风险位置的估算

### 4.2 风险排除计划

## 六、软件质量管理

---



**MONT BLANC Starwalker**



**STABILO**

## 六、软件质量管理

---

### ❖ 1. 软件质量

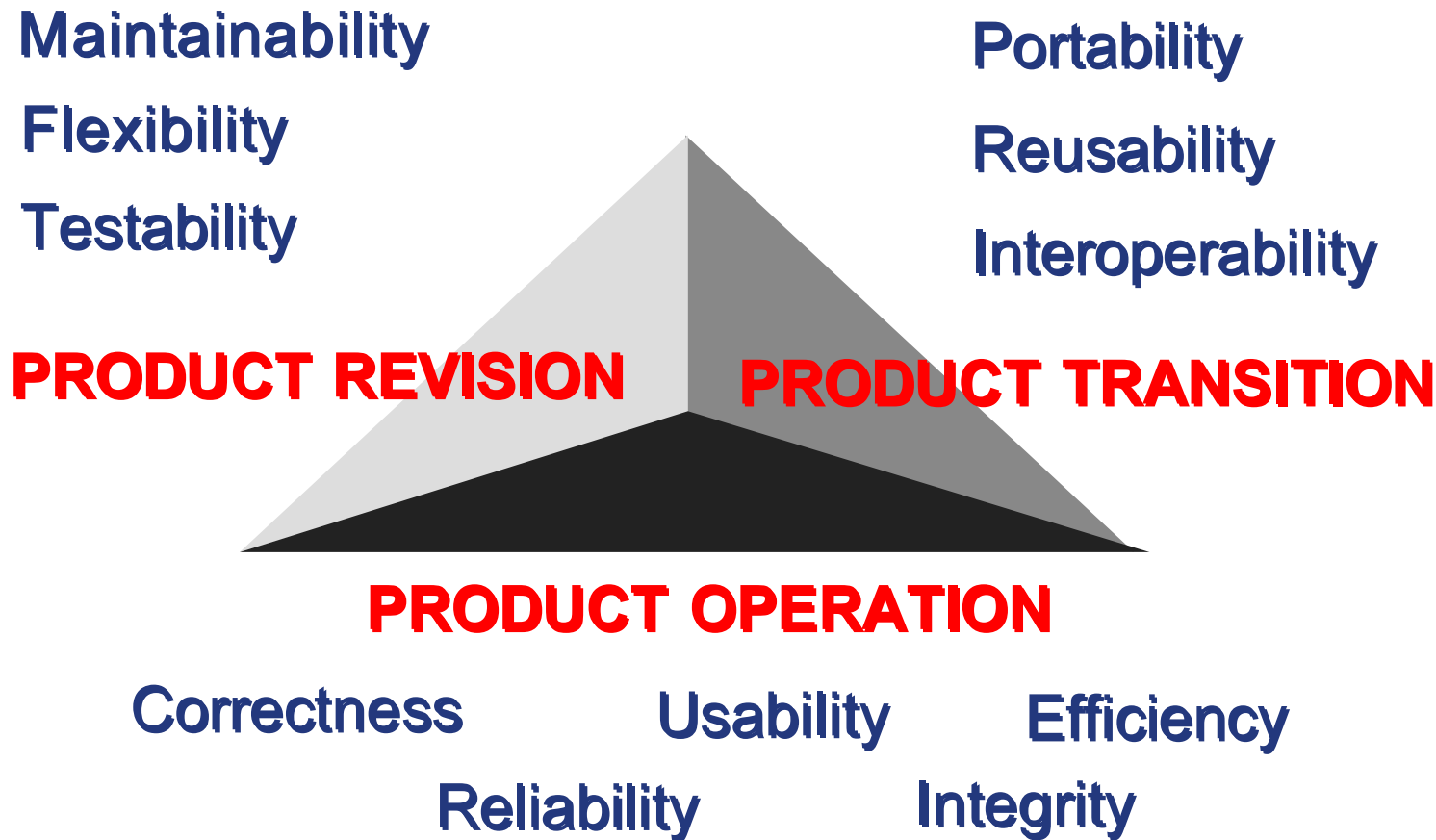
❖ **For software, two kinds of quality may be encountered:**

- **Quality of design** encompasses requirements, specifications, and the design of the system.
- **Quality of conformance** is an issue focused primarily on implementation.
- **user satisfaction = compliant product + good quality + delivery within budget and schedule**

## 六、软件质量管理

---

### McCall's Triangle of Quality



# 六、软件质量管理

---

## ❖ 2. 质量控制

❖ 质量控制是为了保证每一件工作产品都满足对它的需求而应用于整个开发周期中的一系列审查、评审和测试

- 质量控制在创建工作产品的过程中包含一个反馈循环，通过对质量的反馈，使得我们能够在得到的工作产品不能满足其规约时调整开发过程
- 所有工作产品都应该具有定义好的和可度量的规约，这样就可以将每个过程的产品与这一规约进行比较

## 六、软件质量管理

---

### ❖ 3. 质量保证

- 质量保证由管理层的审计和报告构成，目标是和管理层提供获知产品质量信息所需的数据，从而获得产品质量是否符合预定目标的认识和信心



## 六、软件质量管理

---

### ❖ 3. 质量保证

#### ❖ 软件质量保证活动由两类不同的角色承担

- 负责技术工作的软件工程师：通过采用可靠的技术方法和措施、进行正式的技术评审、计划周密的软件测试来考虑质量问题，并完成软件质量保证和质量控制活动
- 负责质量保证工作的**SQA (Software Quality Assurance)**小组：辅助软件工程小组得到高质量的最终产品

# 六、软件质量管理

---

## ❖ 4. SQA小组的活动(CMU SEI)

- 为项目准备**SQA**计划
- 参与开发该项目的软件过程描述
- **评审**各项软件工程活动,核实其是否符合定义好的软件过程中的相应部分
- 审计指定的软件工作产品,并进行核实
- 确保软件工程及工作产品中的偏差已记录在案,并根据预定规程进行处理
- 记录所有不符合的部分并报告给高管
- 此外, **SQA**小组还需要协调变化的控制和管理,并帮助收集和分析软件度量信息

# 六、软件质量管理

---

## ❖ 5. 评审

- 评审是软件质量保证的重要手段
- 评审时机
  - 通常在软件工程过程的每个活动(如需求分析、设计、编码)的后期进行
- 主要评审活动
  - 项目管理评审
  - 技术评审

# 六、软件质量管理

---

## ❖ 5. 评审

### ❖ 1) 项目管理评审

- 任务：针对适用的项目计划、进度安排、标准和指南进行项目状态的评价
- 评审的结果应做出下列规定：
  - 基于对活动或软件产品状态的评价，按照计划进行改进活动
  - 通过配备必要的资源维持项目的总体控制
  - 改变项目的方向或决定是否需要另外计划
  - 评价和管理可能危及项目成功的风险问题

# 六、软件质量管理

---

## ❖ 5. 评审

### ❖ 2) 技术评审

- 任务：评价软件产品或服务，并提供下列证据
  - 完整性
  - 符合标准和规范
  - 其更改是正确地实施的，并且仅仅影响配置管理过程所标明的区域
  - 它们遵循适用的规程
  - 它们已为下一个活动做好准备

# 六、软件质量管理

---

## ❖ 5. 评审

### ❖ 3) 正式评审和非正式评审

#### ■ 正式评审(**formal reviews**)

- 通常在软件工程过程的每个活动的后期进行
- 采用正式的会议评审方式
- 通过正式评审意味着里程碑和基线

#### ■ 非正式评审(**informal reviews**)

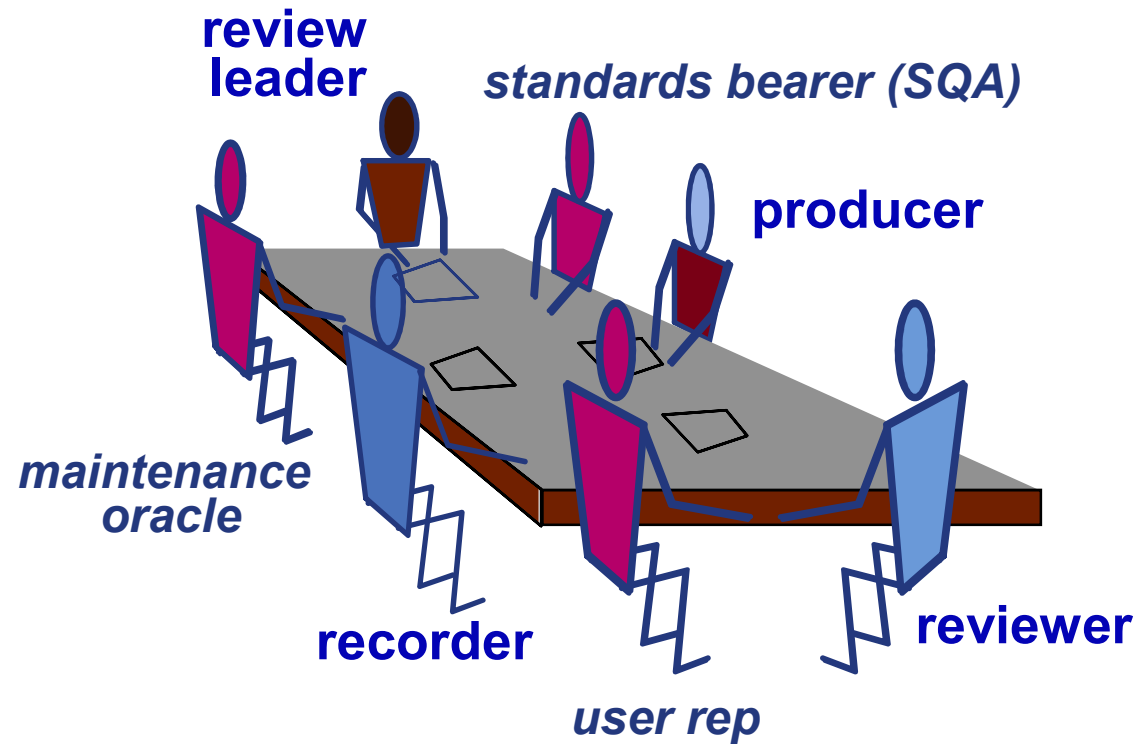
- 由同事参加的即兴聚会
- 可采用“同行评审”(peer review)或“走查”(walkthrough)的方式

# 六、软件质量管理

## ❖ 5. 评审

### ❖ 4) 正式的技术评审过程

- 评审会议



# 六、软件质量管理

---

## ❖ 5. 评审

### ❖ 4) 正式的技术评审过程

#### ■ 评审记录

- 指派专人记录会上提出的所有问题
- 会议结束后将其整理成一份“评审问题列表”并存档

#### ■ 评审报告

- 评审结束后形成报告，报告应指明被评审的制品，参加评审的人员，评审中发现的问题以及评审的结论
- “评审问题列表”可作为评审总结报告的附件。



# 六、软件质量管理

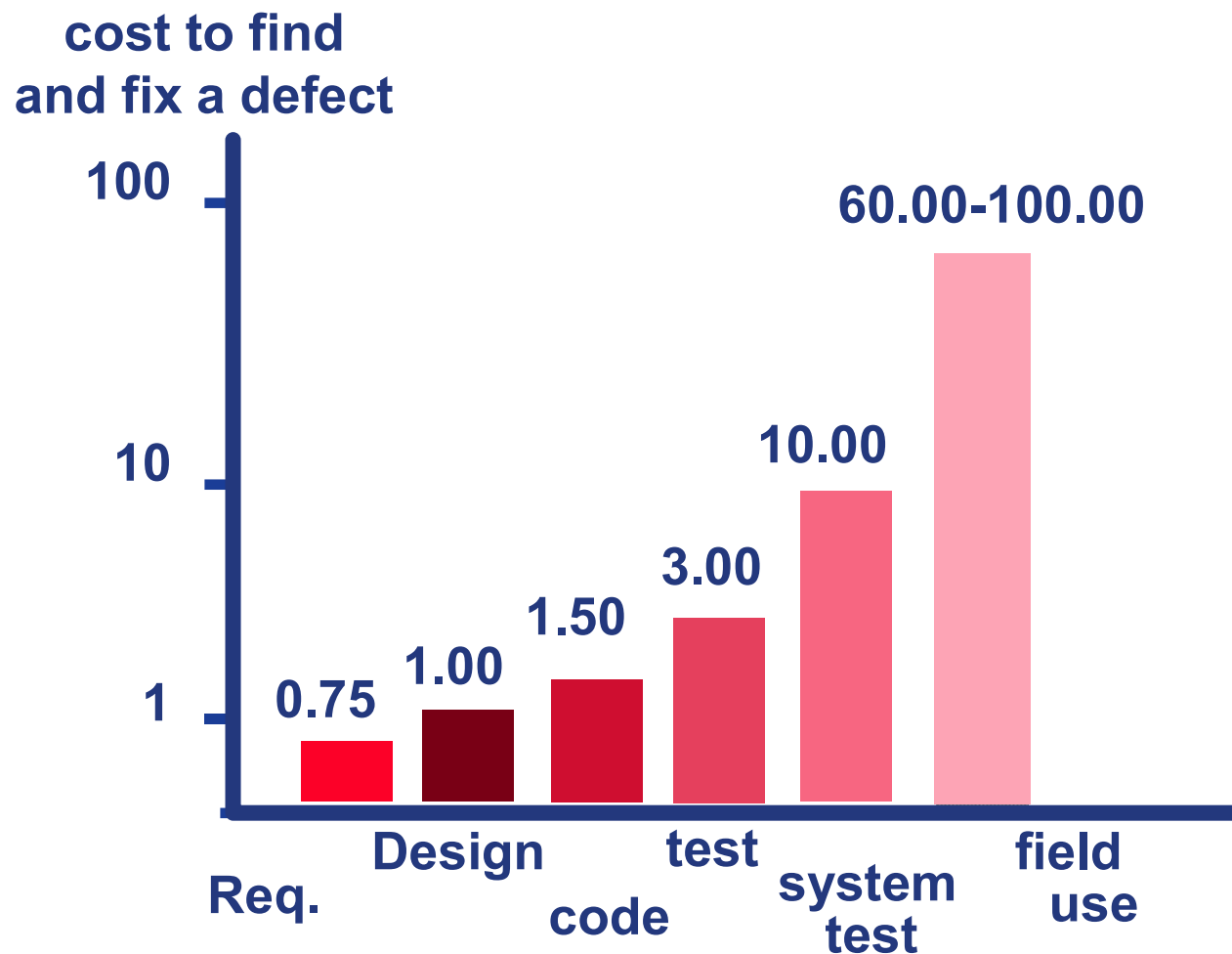
---

## ❖ 5. 评审

### ❖ 5) 评审的指导原则

- 评审产品，而不是评审生产者
- 制定议事日程且遵守日程
- 限制争论和辩驳
- 对各个问题都发表见解，但不要试图解决做书面笔记
- 限制参与者人数并坚持事先做准备
- 为每个可能要评审的工作制品建立一个检查表
- 为正式技术评审分配资源和时间
- 对所有评审者进行有意义的培训
- 评审以前所做的评审

## 六、软件质量管理——6. 质量成本



# 六、软件质量管理——6. 质量成本

---

## ❖ Prevention Costs (COP) 预防成本

- training
- standard, procedures
- planning
- quality improvement
- audits
- analysis

## ❖ Appraisal Costs (COA) 验证成本

- reviews
- walkthroughs
- testing
- supplier monitoring

## 六、软件质量管理——6. 质量成本

---

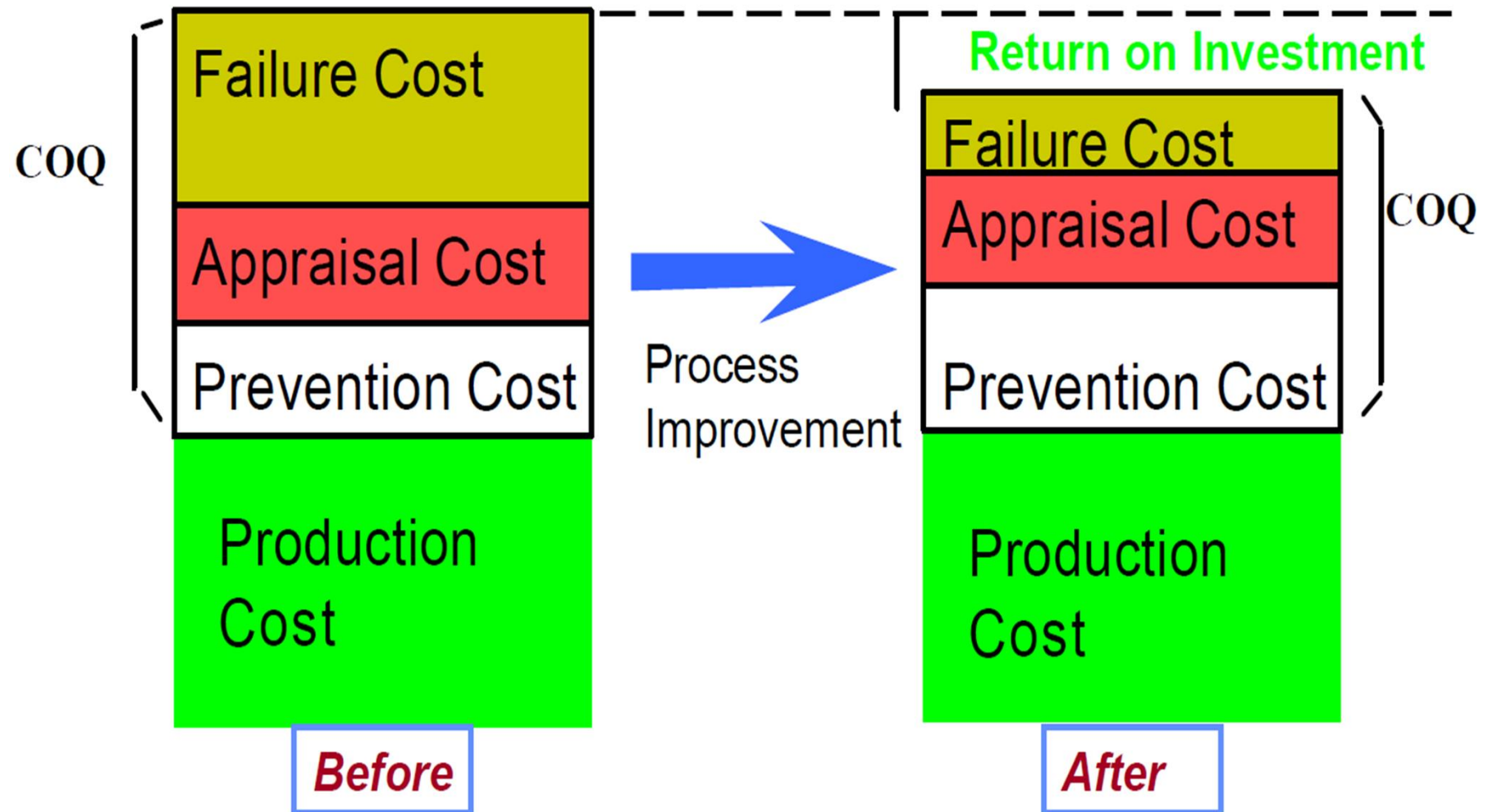
### ❖ Internal Failure Costs 内部失效成本

- Correction and re-work
- Regression testing
- Wasted tester time
- Cost of late shipment
- Opportunity cost of late shipment

### ❖ External Failure Costs 外部失效成本

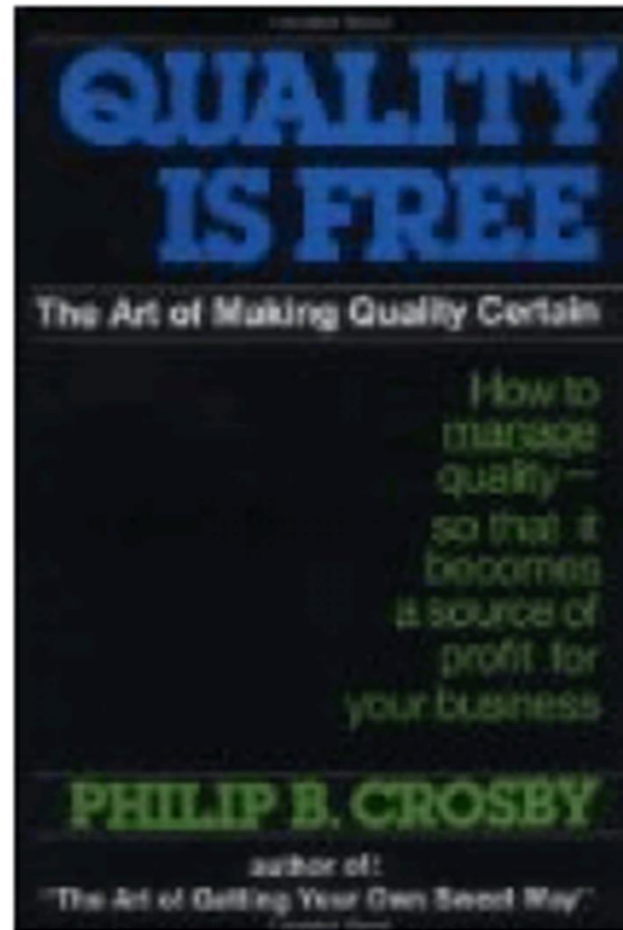
- Technical support calls
- Investigate customer complaints
- Refunds and recalls
- Lost sales
- Lost goodwill
- Penalties
- Liability costs
- Legal fees

## 六、软件质量管理——7. Quality is Free!!



## 六、软件质量管理——7. Quality is Free

---

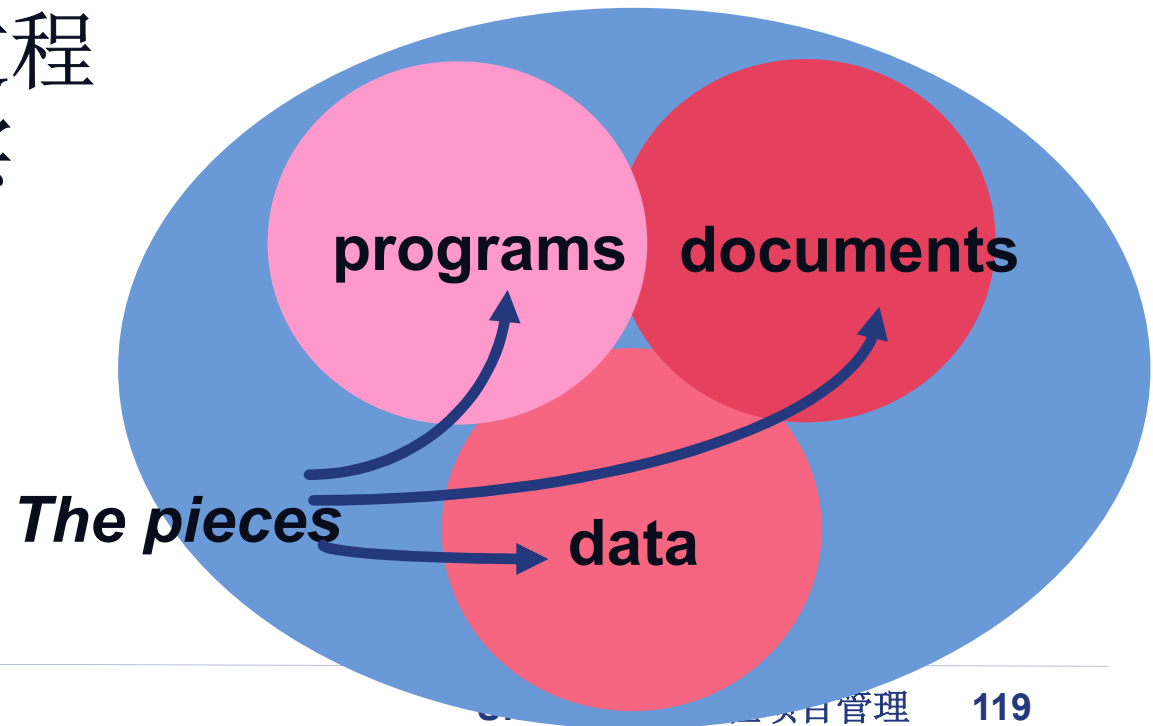


# 七、软件配置管理

## ❖ 1. 相关概念

### ❖ 1) 软件配置项(Software Configuration item, SCI)

- 为配置管理设计的软件工作产品的集合，它在配置管理过程中作为单个实体对待。



# 七、软件配置管理

---

## ❖ 1. 相关概念

## ❖ 2) 配置管理(Configuration management): 应用技术的和管理的指导和监控方法以

- 标识和说明配置项的功能和物理特征
- 控制这些特征的变更
- 记录和报告变更处理和实现状态
- 验证与规定的需求的遵循性



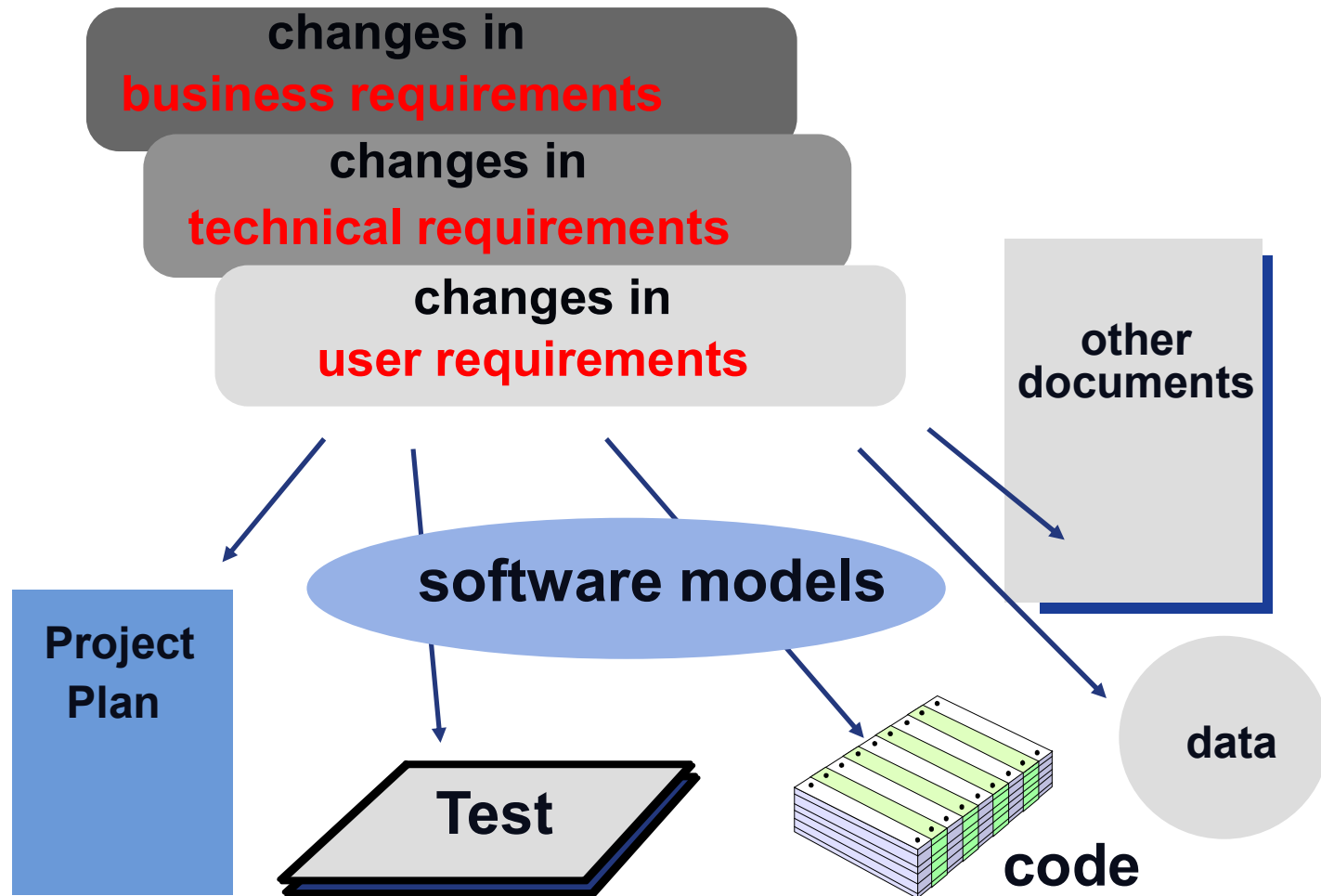
## 七、软件配置管理——3) 变更

---

**No matter where you are in the system life cycle, the system will **change**, and the desire to change it will persist throughout the life cycle.**

*Bersoff, et al, 1980*

## 七、软件配置管理——3) 变更



# 七、软件配置管理

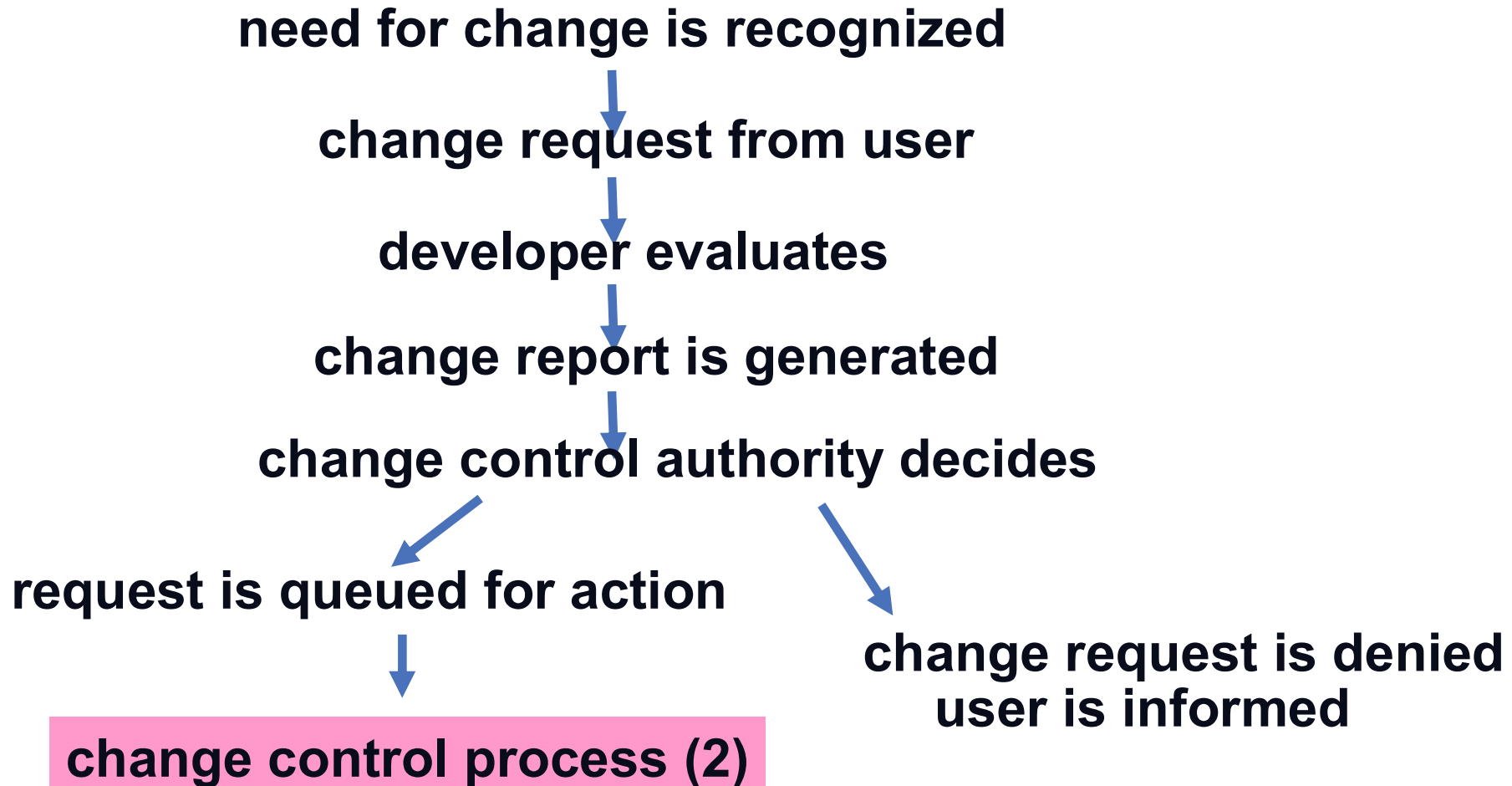
---

## ❖ 1. 相关概念

- **4) 版本(Version):** 与计算机软件配置项的完全编篡或重编篡相关的计算机软件配置项的初始发布或再发布。
- **5) 发布(Release):** 一项配置管理行为, 它说明某配置项的一个特定版本已准备好用于特定的目的(例如发布测试产品)
- **6) 基线(baseline):** 业已经过正式审核与同意, 可用作下一步开发的基础, 并且只有通过正式的修改管理过程方能加以修改的规格说明或产品

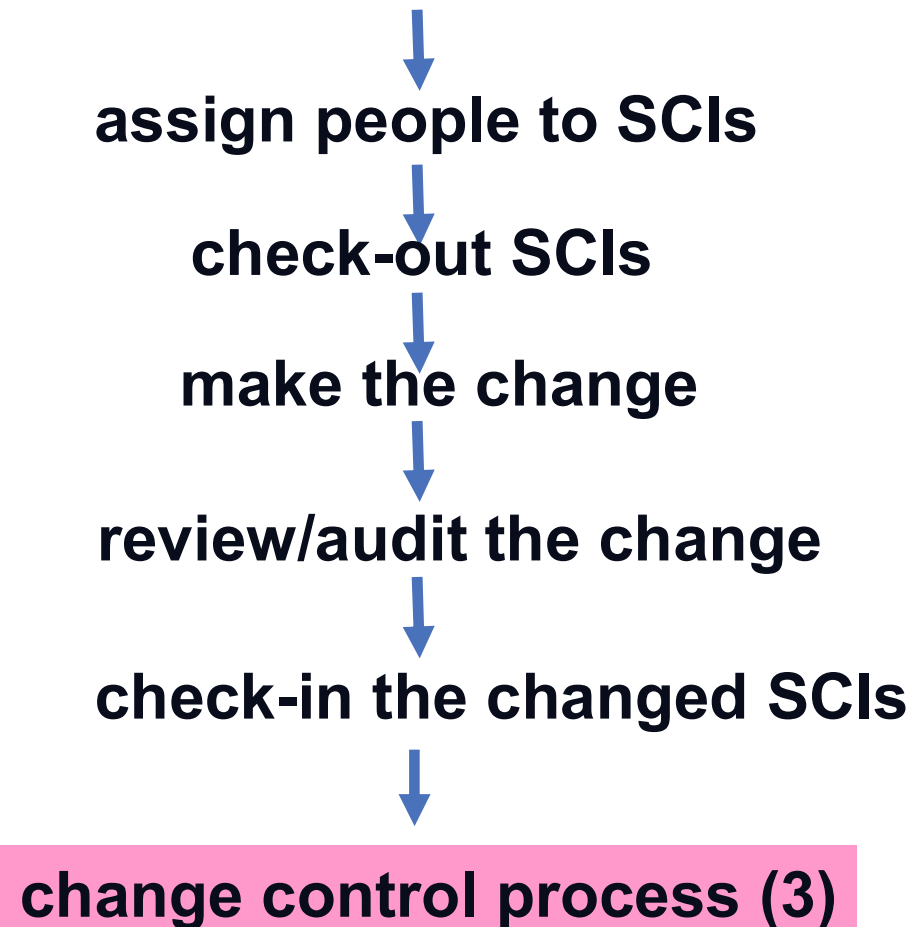
## 七、软件配置管理——2. 变更控制过程

---



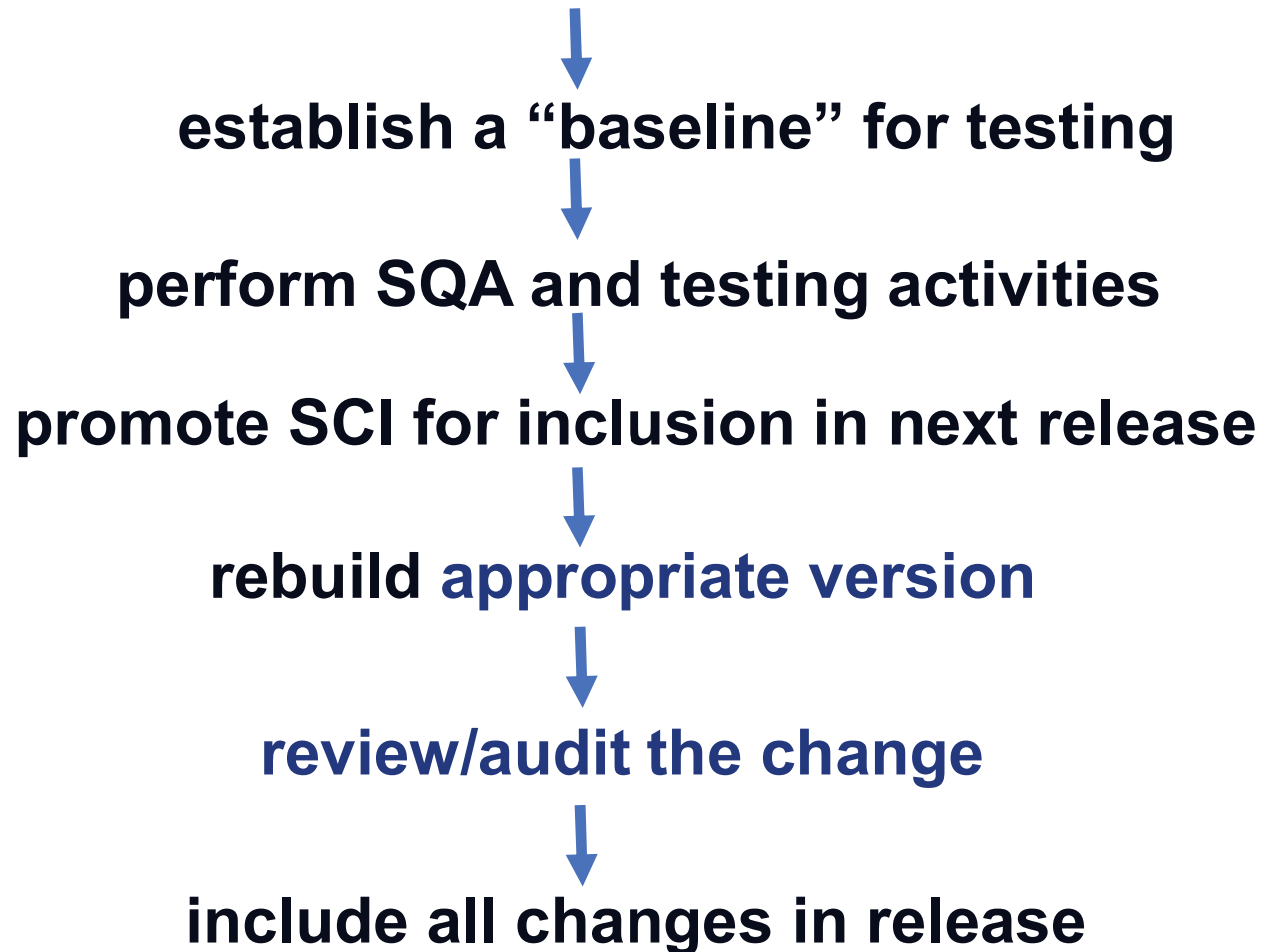
# Change Control Process (2)

---

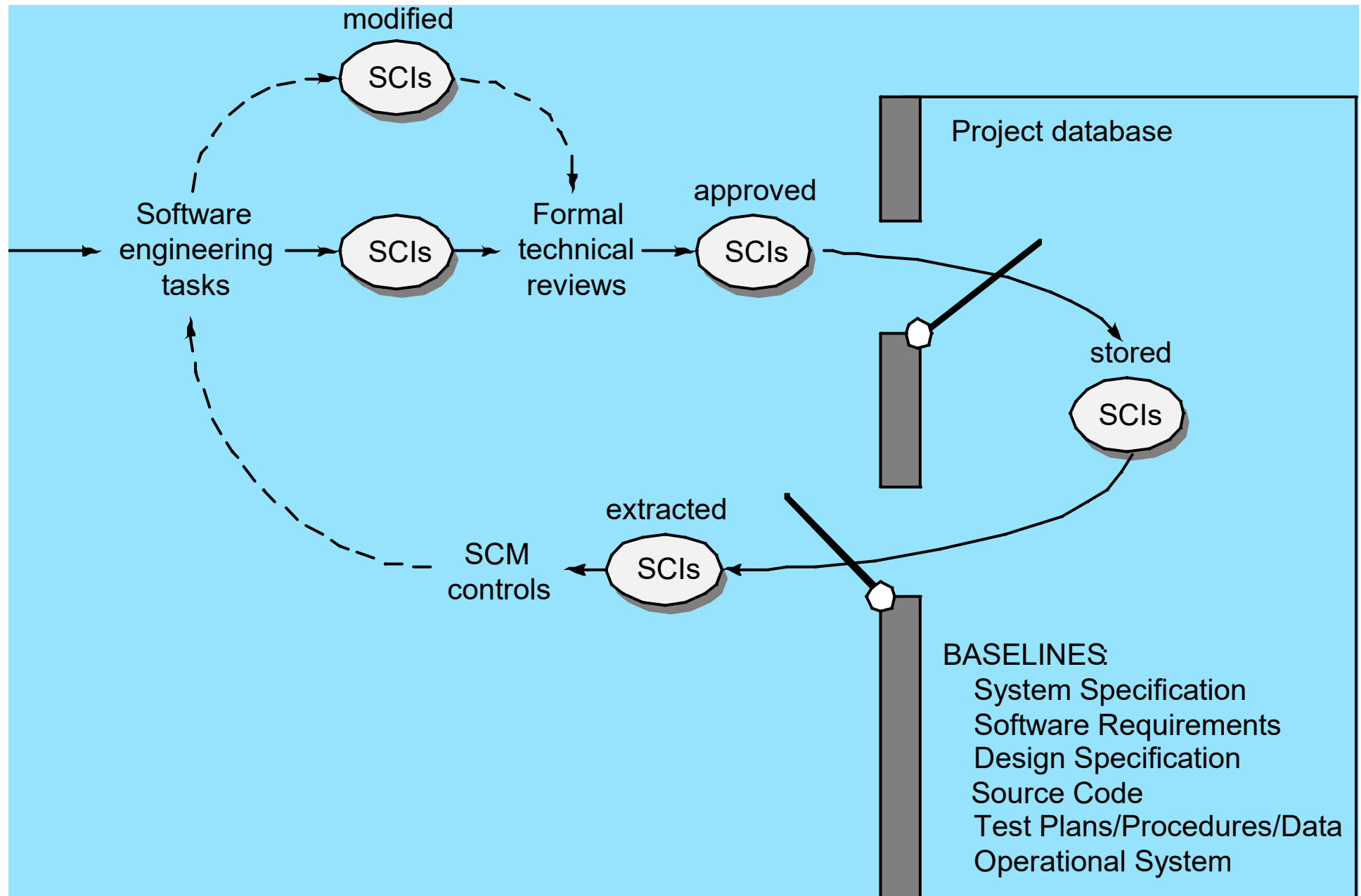


# Change Control Process (3)

---



# 七、软件配置管理——3. 基线的作用



# 七、软件配置管理

---

## ❖ 3. 基线的作用

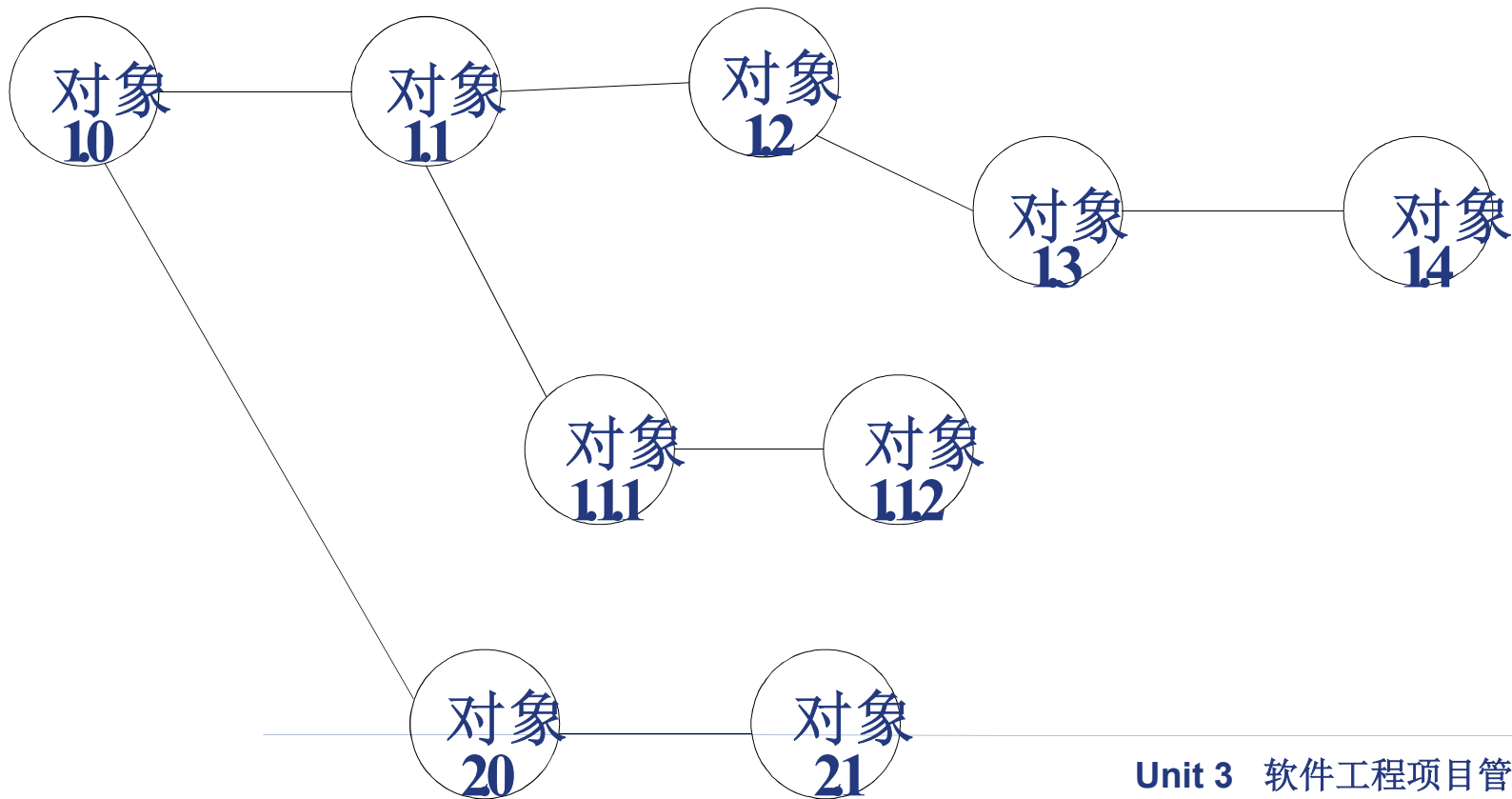
- 在配置项变成基线之前，开发者可以进行非正式的变更控制
- 一旦配置项已经经过正式的技术审评并创建了配置项的基线，则要实施项目级的变更控制
  - 为了对其进行修改，开发者必须获得项目管理者的批准(如果变更是“局部的”)或此配置项的变更授权人的批准(如果该变更影响到其他配置项)



# 七、软件配置管理

## ❖ 4. 版本控制

- 对系统不同版本进行标识和跟踪的过程
- 演化图：描述对象的变更历史

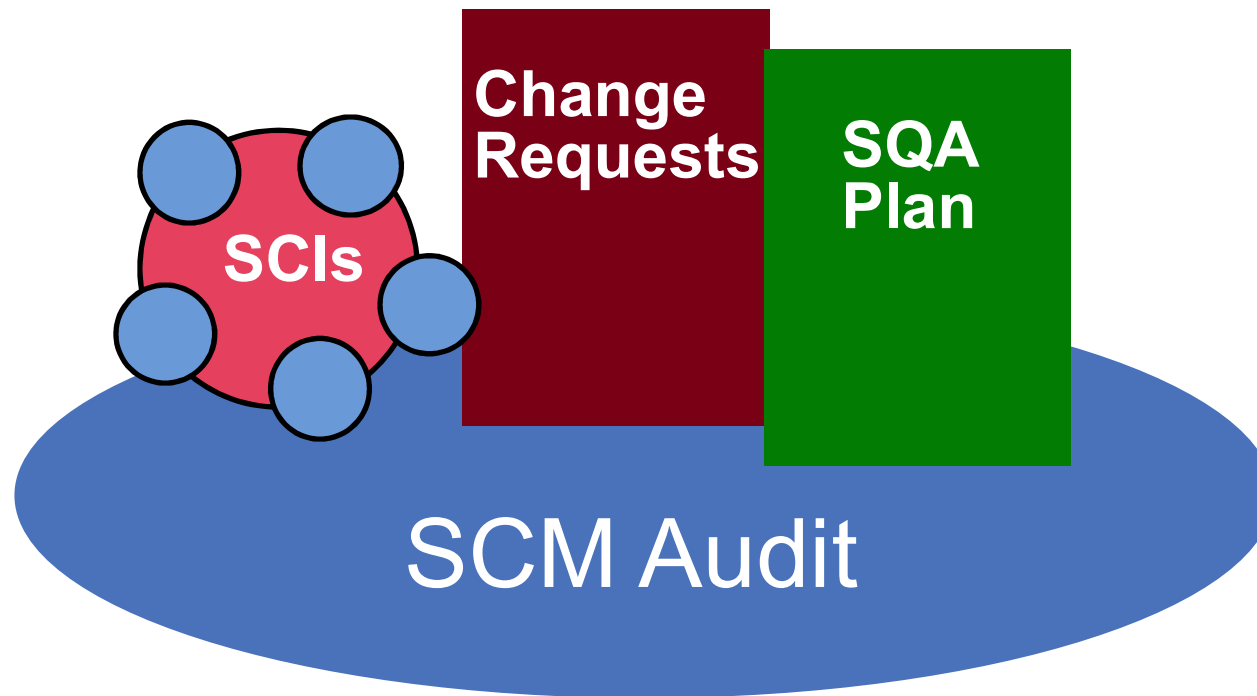


# 七、软件配置管理

---

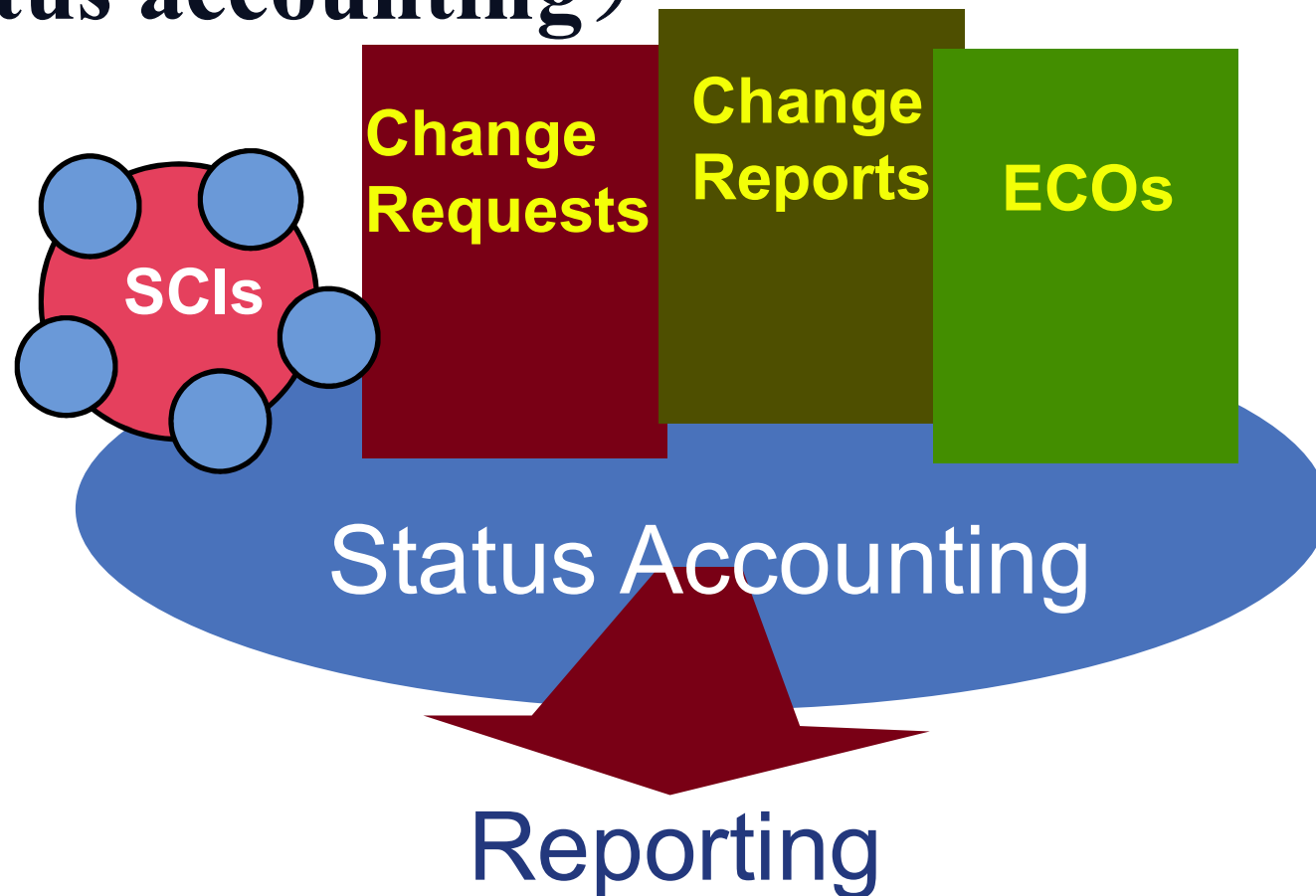
## ❖ 5. 配置审计

- 通过正式技术评审或软件配置审计来保证变更的有效性



## 七、软件配置管理

### ❖ 6. 配置状态报告（也称为状态记录，**status accounting**）





Southeast University



# Thank You !

[lliao@seu.edu.cn](mailto:lliao@seu.edu.cn)