

# 章节重点

## 第一章：绪论

### 1. 软件的定义及特征

**软件：**是使计算机能够工作的指令集合和相应的数据结构和文档，是一种产品，将计算机的硬件能力发挥出来的一种工具，是传递信息的一种工具，对信息的处理手段。

**软件的特征：**

1. 软件是一种逻辑元素，而不是物理元素；
2. 软件是开发出来的，而不是用传统的方法制造出来的；
3. 软件不会被用坏，一般产品的失败概率都遵循浴盆曲线；
4. 工业界已经是标准化装配时代，但软件还是定制时代；
5. 创新性和人为因素更高。

### 2. 软件危机产生的原因

- 用户不易准确描述对软件的需求，经常存在二义性，遗漏甚至错误
- 大型软件往往需要成百上千人的合作，由于软件系统结构复杂，如何有效组织管理、充分发挥团队作用就成为软件开发成功的关键。
- 缺乏有效的软件开发方法和工具的支持，过分依靠程序设计在开发中的技巧和创造性，加剧了软件产品的个性化。开发过程没有统一、规范的方法论指导，文档资料不齐全。
- 缺乏软件开发经验及相关数据积累，无法准确估计经费和进度，导致经费严重超支，完成期限一拖再拖。
- 忽视测试阶段的工作，提交的产品质量差。

### 3. 软件过程、软件过程能力、软件过程性能的定义

**软件过程：**人们用于开发和维护软件及其相关过程的一系列活动，包括软件工程活动和软件管理活动。

**软件过程能力：**描述（开发组织或项目组）遵循其软件过程能够实现预期结果的程度，它既可对整个软件开发组织而言，也可对一个软件项目而言。

**软件过程性能：**表示（开发组织或项目组）遵循其软件过程所得到的实际结果，软件过程性能描述的是已得到的实际结果，而软件过程能力则描述的是最可能的预期结果，它既可对整个软件开发组织而言，也可对一个特定项目而言。

### 4. CMM 定义及 5 个成熟度级别的主要特征

**SEI 给 CMM 下的定义是：**对于软件组织在定义，实现，度量，控制和改善其软件过程的进程中各个发展阶段的描述。

- **初始级特征：**软件过程的特点是无秩序的，偶尔甚至是混乱的，几乎没有什么过程是经过定义的，成功依赖于个人努力。
- **可重复级特征：**已建立基本的项目管理过程去跟踪成本进度和功能，必要的过程纪律已经就位，使具有类似应用的项目能重复以前的成功。
- **已定义级特征：**管理活动和工程活动两方面的软件过程均已文档化、标准化，并集成到组织的标准软件过程中，全部项目均采用供开发和维护软件用的组织标准软件过程的一个经批准的普及剪裁版本。
- **已管理级特征：**已采集详细的有关软件过程和产品质量的度量，无论软件过程还是产品均得到定量了解和控制。

- **优化级特征:** 利用来自过程和来自新思想、新技术的先导性实验的定量反馈信息, 使持续过程的改进成为可能。

## 5. 软件过程的可视性、过程能力和性能预测与成熟度之间的关系

- 随着成熟度增长, 可视性越来越高
- 随着成熟度增长, 实际结果相对预定目标结果的偏差范围减小
- 随着成熟度增加, 预定目标结果得到改善

## 6. KPA 定义及结构

**关键过程区域 (Key Process Area, KPA):** 每个关键过程区域识别出一串相关活动, 当这些活动全部完成时, 能达到一组对增强过程能力至关重要的目标。

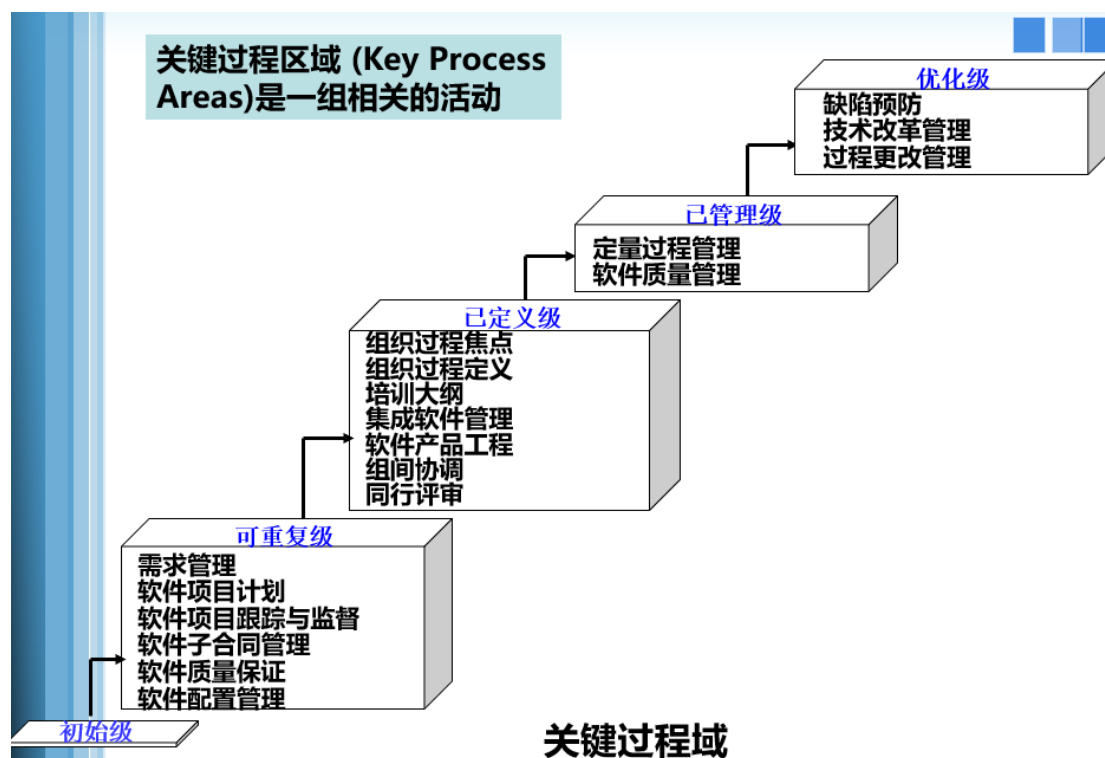
**KPA 的结构:**

- 目标
- 共同特点: 执行约定、执行能力、执行活动、度量和分析、验证实施

## 7. KPA 的共同特点包含 5 点内容

- **执行约定:** 企业为了建立和实施相应 KPA 所必须采取的行动;
- **执行能力:** 描述了为了某软件过程得以始终如一地执行必须在项目或企业中存在的先决条件, 是企业实施 KPA 的前提条件;
- **执行活动:** 描述了执行 KPA 所需求的必要行动、任务和步骤; 其是唯一一项与项目执行相关的属性。
- **度量和分析:** 关注于这个关键过程域的活动需要做的度量和度量分析要求。
- **验证实施:** 是验证执行活动是否与建立的过程一致, 核实以确保所实施的过程是按照原定的计划以及达到其目标, 着眼于保证过程的实现要通过独立的个人和高级管理人员验证。

## 8. CMM 每个成熟度等级的 KPA 及其子目标



1. 初始级
2. 可重复级 (6 个)

- **需求管理**
  - ①软件需求受控制，为软件工程和管理活动建立基线；②软件计划、产品和活动与需求保持一致。
- **软件项目计划**
  - ①项目活动和约定是有计划的，并成文归档；②受影响的小组和个人认同项目相关的约定；③对项目的估计要文档化，便于在项目计划和跟踪时使用。
- **软件项目跟踪与监督**
  - ①实际的结果和性能要按照软件计划进行跟踪；②实际的结果和性能严重偏离软件计划时，要采取正确的措施，并设法关闭；③受影响的组和个人认同对约定的更改
- **软件子合同管理**
  - ①主合同方和子合同方认同他们的约定；②主合同方根据他们的约定跟踪子合同方的实际结果；③主合同方和子合同方在项目进行期间保持联系；④主合同方根据他们的约定跟踪子合同方的实际性能。
- **软件质量保证**
  - ①软件质量保证活动纳入计划；②软件产品和活动与采用的标准、规程和需求间的一致性被客观验证；③受影响的组和个人知道软件质量保证活动和结果；④项目中无法解决的有意见分歧的事宜通知高级管理部门处理。
- **软件配置管理**
  - ①软件配置管理活动纳入计划；②选定的软件工作产品被标识、受控制和便于利用；③被标识的软件工作产品的更改是可控的；④受影响的小组和个人知道软件基线的状态和内容。

### 3. 已定义级（7个）

- **组织过程焦点**
  - ①软件过程开发和改进活动在机构内保持一致；②所使用的软件过程的强弱之处被确定；③机构级的过程开发和改进活动纳入计划。
- **组织过程定义**
  - ①开发并维护机构的一个标准软件过程；②与软件项目使用机构标准软件过程相关的信息被收集、评审并可用。
- **培训大纲**
  - ①培训活动纳入计划；②提供关于执行软件管理和技术的人员所需的开发技巧和知识的培训；③软件工程组和软件相关组里每个人都接受了执行工作所需的培训。
- **集成软件管理**
  - ①项目定义的软件过程是机构标准软件过程的裁剪版本；②依据项目定义的软件过程来计划和管理项目。
- **软件产品工程**
  - ①软件工程任务被定义、集成并遵照执行，以生产软件；②软件工作产品相互间保持一致。
- **组间协调**
  - ①所有受影响的组认同客户需求；②所有的组认同不同组间的约定；③各组识别、跟踪和解决组间问题。
- **同行评审**
  - ①同行评审活动纳入计划；②软件工作产品的缺陷被识别和清除。

#### 4. 已管理级（2个）

- **定量过程管理**

①将量化过程管理活动纳入计划；②项目定义的软件过程的过程性能得到量化控制；③机构标准软件过程的过程能力以量化项值表示。

- **软件质量管理**

①将项目的软件质量管理活动纳入计划；②对软件产品质量及其优先级的可测目标有定义；③软件产品质量目标的实际进步得到量化管理。

#### 5. 优化级（3个）

- **缺陷预防**

①将缺陷预防活动纳入计划；②找出和识别引起缺陷的共同原因；③引起缺陷的共同原因被赋予优先次序并被系统地清除。

- **技术改革管理**

①将技术更新纳入计划；②新技术被评估以决定对质量和生产率的影响；③合适的新技术被引入到机构内正常使用。

- **过程更改管理**

①将持续地过程改进纳入计划；②机构软件过程改进活动的参加者遍及全部机构；③机构软件过程和项目定义的软件过程得到持续地改进。

## 第二章：软件项目管理概述

### 1. 项目的定义和特征

**项目：**所谓项目，就是为创建某一独特产品或服务或成果，在一定的环境和约束条件下进行的临时性努力。即它是利用有限的资源，在有限的时间内为特定客户完成特定目标的一次性工作。

**项目的特征：**

1. 一个明确的范围和目标；
2. 一个预期的完成时间；
3. 有可以利用的资源；
4. 一种已定义的性能评估方法；
5. 不是例行的任务

### 2. 项目管理的定义

一定的主体，为了实现其目标，综合运用专门的知识、技能、工具和方法，对执行中的项目周期的各阶段工作进行计划、组织、协调、控制，以满足甚至超越项目干系人的需求和期望。

### 3. 项目管理的知识体系

1. **范围管理：**按照某个特定目标，确定和控制某个项目范围的过程；
2. **质量管理：**在遵循客户需求和期望下，确保项目性能符合规范；
3. **时间管理：**有效利用时间，以方便执行项目。主要以项目规划的方式实现；
4. **成本管理：**通过运行有关评估预测预算和报告的可靠技术使我们能有效地控制成本；
5. **风险管理：**识别、分析并确认各种可能影响项目的风险和不确定因素的过程；
6. **人力资源管理：**在项目生命周期内，对人力资源进行有效管理；
7. **合同/采购管理：**获取必要的设备、工具、服务、物品和资源以成功实现项目目标；
8. **通讯管理：**项目环境中个人和团体之间的功能结合，包括恰当的组织、传递和控制为促进工作所需要的信息；
9. **项目综合管理：**综合上述管理内容，使项目得以有效进行。

#### 4. 项目管理的三要素

质量、进度、成本

#### 5. 项目管理的阶段划分

- **项目规划**：主要是项目经理审阅合同条款，并制定一个满足他们的计划，实际上包括：定义生命周期、估计工作量和进度、制定任务进度计划等。
- **项目执行**：包括执行项目计划、跟踪项目的状态，并在项目的绩效偏离项目计划设定的绩效时采取措施进行纠正。
- **项目收尾**：主要是在客户接收工作产品之后对项目进行系统的总结。数据分析是这一阶段的主要任务。

### 第三章：需求分析

#### 1. 需求的定义

**IEEE 软件工程标准词汇表（1997 年）定义需求为：**①用户解决问题或达到目标所需的条件或权能。②系统或系统部件要满足合同、标准、规范或其它正式规定文档所需具有的条件或权能。③一种反映上面①或②所描述的条件或权能的文档说明。

#### 2. 需求分析的过程

1. **准备阶段**：阅读技术以及商务概念上的背景资料并进行培训、熟悉客户使用的方法和工具、确定信息的采集方法、准备好提问问题、确定用户组与评审专家、计划原型、确定需求规格标准、制定会谈计划；
2. **采集、澄清需求**：建立系统目标和范围、采集功能需求、采集外部接口信息、采集环境需求、采集性能需求、采集标准需求、采集用户特殊需求、准备和评估原型；
3. **分析需求**：设计过程模型、设计逻辑数据模型、建立数据字典；
4. **准备 SRS**
5. **评审 SRS**
6. **客户认可并签署 SRS**

#### 3. 需求规格说明书的要求

- **正确性**：与分配需求相吻合，满足客户期望；
- **无二义性**：任何内容的陈述只能有一种确定的形式；
- **完整性**：描述所有有意义的需求；
- **一致性**：不存在有矛盾的需求；
- **可测试性**：任何需求应可以测试，即存在有限代价的有效过程验证确认；
- **可跟踪性**：每条需求的来源、流向是清晰的，当产生和改变文件编制时，可以方便地引证每一条需求。

#### 4. 需求变更管理的过程

- 记录变更
- 分析变更对工作产品的影响
- 估计变更申请所需的工作量
- 重新估计交付时间表
- 执行累计的成本影响风险
- 如果影响超出一定的限度，则与高级主管一起评审影响
- 客户不再提出变更申请
- 修改工作产品

#### 5. 进行需求跟踪管理常用的工具是什么？

**跟踪矩阵**：支持跟踪功能最简单的方式是进行从需求单元到设计单元、从设计单元到编

码单元、从编码单元到测试用例的映射。

## 第四章：过程定义和过程裁剪

### 1. 过程的定义

**过程：**过程描述是项目可以用来遵照执行某些任务的一系列步骤，以及执行这些步骤的指南。开发过程是提炼用户需求，设计、构建和测试满足这些需求的软件，并最终将其交付给客户所需的过程。当开发新的应用或计划为现有的应用进行重要的增强时，就要使用开发过程。

### 2. 一般的软件开发过程包括哪些子过程

1. 需求分析
2. 概要设计
3. 详细设计
4. 编码和单元测试
5. 集成测试
6. 系统测试
7. 文档
8. 验收测试和安装
9. 系统维护

### 3. 软件开发每个子过程的参加者、五要素

**五要素：**输入准则、输入、输出准则、输出、度量

1. **概要设计：**给出从计算机的逻辑角度开发针对用户需求的解决方案。
  - 参加者：设计团队、评审团队、客户
  - 输入准则：需求规格文档经过评审并授权
  - 输入：需求规格文档
  - 输出准则：概要设计文档经过评审和授权
  - 输出：概要设计文档、项目标准、概要设计评审记录
  - 度量：概要设计工作量、概要设计缺陷、评审工作量、返工工作量
2. **详细设计：**进一步对概要设计中的整体应用分解，分解成模块和程序，对程序进行逻辑设计。
  - 参加者：设计团队
  - 输入准则：概要设计文档经过评审和授权
  - 输入：概要设计文档
  - 输出准则：详细设计文档和单元测试计划已经经过评审和授权
  - 输出：详细设计文档和单元测试计划
  - 度量：详细设计工作量、设计缺陷、单元测试缺陷、程序框架缺陷、评审和返工工作量
3. **编码和单元测试：**根据详细设计用编程语言编写所需要的程序
  - 参加者：项目组成员、项目经理
  - 输入准则：详细设计文档经过评审并授权
  - 输入：详细设计文档、项目标准、程序规格、单元测试计划、程序框架、可利用开发的实用程序工具及文档
  - 输出准则：成功执行所有单元测试计划中的测试用例
  - 输出：源代码、可执行代码、测试数据

- 度量：编码和单元测试的工作量、代码评审缺陷、独立单元测试缺陷、评审和返工工作量
- 4. **集成测试：**已通过单元测试的模块构建成一个完整软件结构的系统方法
  - 参加人员：集成测试团队
  - 输入准则：概要设计文档经过评审和授权
  - 输入：概要设计文档和程序
  - 输出准则：成功执行所有集成测试计划中的测试用例
  - 输出：源代码、可执行代码、测试数据
  - 度量：工作量、缺陷
- 5. **系统测试：**依据需求规格验证软件产品有效性的活动；目的是发现那些只有通过测试整个系统才能暴露的缺陷
  - 参加者：系统测试团队
  - 输入准则：需求规格和概要设计文档经过评审和授权
  - 输入：需求规格和概要设计文档
  - 输出准则：成功执行所有系统测试计划中的测试用例
  - 输出：源代码、可执行代码、测试数据
  - 度量：工作量、缺陷
- 6. **文档：**主要是操作手册，用户手册及客户需要的其他文档。
- 7. **验收测试和安装：**把软件产品集成到它的操作环境中，并在这个环境中经受测试，确保它按需求执行。
  - 参加者：安装团队、客户、项目经理
  - 输入准则：成功的完成系统测试
  - 输入：测试后的软件和验收测试文档
  - 输出准则：客户签署验收单
  - 输出：安装后的软件
  - 度量：工作量和缺陷
- 8. **系统维护：**
  - 参加者：安装团队、维护团队
  - 输入准则：在运行的软件
  - 输入：安装后的软件、用户文档和软件故障报告
  - 输出准则：合同中指定的维护支持阶段终止及客户签署整个项目
  - 输出：客户签署的文档、已经安装的软件
  - 度量：工作量、缺陷

#### 4. 了解软件开发每个子过程的步骤

1. **需求分析：**初步沟通、到处需求、分析和精化、可行性研究、协商与沟通、规格说明、需求验证、变更管理
2. **概要设计：**
  1. 定义标准，包括编码标准、文档标准和用户接口标准等等
  2. 确定操作环境
  3. 进行功能设计：确定功能。确定商务事件和功能之间的交叉参考。确定每个功能的处理细节。生成功能体系结构并归档
  4. 物理数据库设计：确定数据模型。将实体和关系转化为表和记录类型。执行规模估计：对数据进行规模的估计。执行受控非标准化。确定索引
3. **详细设计：**

1. 将功能分成小的组件
2. 如果需要，开发数据移植程序
3. 设计并开发代码框架
4. 开发例程和工具
5. 程序设计：确定程序调用的方法。确定输入和输出。设计程序逻辑。确定数据结构。确定使用的通用例程。写出程序规格并进行评审
6. 计划单元测试：确定单元测试环境。确定单元测试用例。确定测试数据。写出单元测试计划并进行评审
4. **编码和单元测试：**
  1. 生成测试数据库
  2. 生成代码：对程序进行编码。代码评审。记录和修正评审的缺陷
  3. 进行自我单元测试：进行自我单元测试。进行附加测试。更正缺陷
  4. 进行独立单元测试：准备独立单元测试。进行独立单元测试。记录所有缺陷。关闭独立单元测试
5. **集成测试：**
  1. 确定环境需求（硬件、软件、通信、使用模式）
  2. 决定集成规程：确定将要集成的关键模块。确定集成顺序。确定需要测试的接口
  3. 开发集成测试计划：确定测试用例和执行用例的规程。确定测试数据。确定期望输出。重排测试用例
6. **系统测试：**
  1. 确定所需测试环境
  2. 决定系统测试规程：确定测试特性。确定不需要测试的重要特性以及不测试的原因。确定关键测试
  3. 开发测试用例：确定每个测试用例及执行它的规程。确定输入、输出数据需求。确定预期结果
7. **文档：**
  1. 准备用户手册
  2. 准备操作手册
  3. 准备数据转换手册
  4. 准备联机帮助
  5. 评审文档/手册
8. **验收测试和安装：**
  1. 执行验收
    - 1) 为验收制定计划：从客户那里获得验收环境的细节。准备安装计划。准备软件发布文档
    - 2) 参加验收：在验收环境下安装软件。进行实况运行。协助客户进行验收测试。修正验收缺陷。更新文档，以反映所有的变更。获得用户的验收签字
  2. 执行安装
    - 1) 在产品环境下安装软件
    - 2) 搭建产品环境
    - 3) 载入软件和数据
    - 4) 进行实况运行
    - 5) 获得客户对每个安装的签署



- 6) 修改安装缺陷
- 7) 执行用户的培训

## 5. 过程裁剪的定义和分类

**过程裁剪：**是调整组织标准过程的过程，以此来获得用于项目的特定业务或技术需要的过程。

**分类：**概要裁剪指南、详细裁剪指南。

## 6. 概要裁剪可依据的项目特征

1. 团队和项目经理的经验和熟练程度
2. 团队人数最多时的人数
3. 需求透明度
4. 项目持续时间
5. 应用的关键程度

## 7. 详细裁剪可选择的裁减属性

执行、文档化、评审和详细级别。

# 第五章：过程数据库和过程能力基线

## 1. 软件度量的含义、作用

**软件度量：**可以来量化地描述软件过程和软件产品的不同方面的特点。

**作用：**项目计划、控制项目过程、分析和改进组织软件过程

## 2. 过程数据库定义、构成及含义

**过程数据库（PDB）：**是存放从项目可获得的过程性能数据的数据库，这些数据可以用于项目计划、估计、生产率和质量分析等。

**PDB 的构成：**

- **项目特征：**项目特征信息包含项目名称、项目经理和模块经理的名字，项目所属业务单元、项目中使用的过程、应用领域、硬件平台、所用语言、所用 DBMS、项目目标的简短描述、项目风险信息、项目持续时间和团队规模。
- **项目进度：**进度信息主要包含项目期望开始和结束的日期以及实际开始和结束的日期
- **项目工作量：**项目工作量信息包含初始估计工作量和实际总的工作量信息。
- **项目规模：**规模信息包括已经开发的项目的软件规模。按照 LOC（代码行数）、简单、中等或复杂程序的数目以及他们的组合来描述软件规模。
- **缺陷（故障）：**缺陷信息包括在各种缺陷检测活动中发现的缺陷数。
- **风险：**在 PDB 中记录了一些注释，包括对估计的注释和对风险管理的注释。

## 3. PDB 的建立及访问权限

- PDB 由 SEPG 建立
- 项目经理可以阅读

## 4. 过程能力基线的定义、数据及含义（应掌握 PCB 数据项的计算方法）

**过程能力基线（PCB）：**PCB 代表的是按照量化术语描述的过程能力。

**PCB 中的数据：**已交付软件的质量、生产率、进度计划、工作量分布、缺陷注入率、过程中的缺陷清除率、质量成本、缺陷分布

**生产率：**是每人月所生产的功能点。

**质量：**是每个功能点所提交的缺陷数，提交的缺陷数是根据在整个验收测试、安装和维护期间所发现的所有缺陷估算出来的。

**缺陷注入率：**是在项目的生命周期内所有注入的缺陷除以规模得到的值。项目中全部的缺陷数是在过程中各个不同检测阶段所发现的缺陷数和在提交后发现的缺陷数的总和。

**整体缺陷清除率：**是在过程内通过各种缺陷检测活动所发现的缺陷数占全部缺陷数的百分比。

F = 用功能点描述的软件规模

E = 项目花费的所有工作量

D1 = 在验收测试前发现的所有缺陷数

D2 = 验收测试后发现的缺陷总数

$D = D1 + D2$

生产率 =  $F/E$

质量 =  $D2/F$

缺陷注入率 =  $D/F$

过程中的缺陷清除率 =  $D1/D$

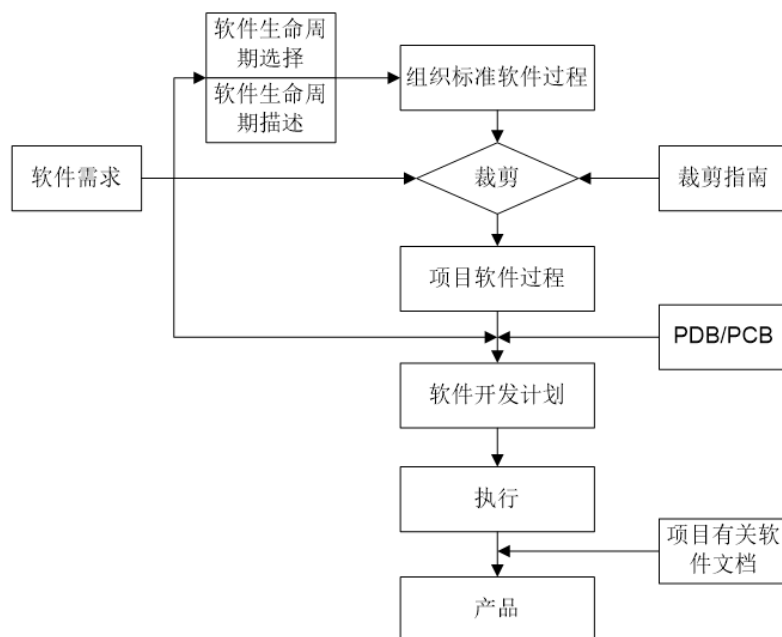
## 5. 影响质量成本的活动包括哪些

- 测试
- 评审
- 软件缺陷修复
- 培训

## 6. 过程财富的组成

- 组织标准软件过程
- 组织的软件过程数据库/过程能力基线
- 软件生命周期描述
- 标准软件过程的剪裁指南和准则
- 项目软件有关文档

## 7. 如何利用过程财富进行软件项目开发



# 第六章：工作量估计和进度安排

## 1. 软件规模估计的方法

- 代码行 (LOC/KLOC) 法
- 功能点法

## 2. 功能点法估计软件规模的步骤 (能够根据案例数据计算估计软件规模)

(例子见功能点和 CoCoMo 的 ppt)

A. 计算未调整的功能点 UFP (unadjusted function points)

确定所需的输入、输出、查询、数据文件和界面等 5 类系统元素的数量和复杂度，然后根据这些数量和复杂度计算各条目的功能点和整个软件的功能点。

B. 计算技术复杂度因子 TCF (Technical Complexity Factor)

通过对 14 个因子的评估，并将各评估值相加，即可获得技术复杂度因子 TCF

$$\in [0,70]: TCF = F1 + F2 + \dots + F14$$

技术因子有:

F1: 可靠的备份和恢复

F8: 联机更新

F2: 数据通信

F9: 接口复杂度

F3: 分布式功能

F10: 可复用性

F4: 性能要求

F11: 处理复杂性

F5: 操作环境

F12: 安装的方便性

F6: 联机数据输入

F13: 多站点

F7: 多屏输入

F14: 使用方便性

取值范围:

0: 不需要或没有影响

3: 一般影响

1: 偶有影响

4: 较大影响

2: 轻微影响

5: 严重影响

C. 计算功能点

$$FP = UFP * (0.65 + 0.01 * TCF)$$

D. 功能点与代码行的转换

编程语言	LOC/FP
C	128
Cobol	105
FORTRAN	105
PASCAL	90
Ada	70
4GL	25
代码生成器	15
电子表格	6

### 3. 自底向上的工作量估计方法步骤 (能够根据案例数据计算估计工作量)

1. 确定系统中的程序，并将它们分为简单、中等复杂或复杂(S/M/C)。尽量使用提供的定义或其他项目使用的定义。
2. 如果存在项目特定的基线，从基线中获得构建 S/M/C 程序所需的平均工作量。
3. 如果项目特性的基线不存在，使用项目类型、技术、语言和其他属性来查找过程数据库中类似的项目。使用来自这些项目的数据来定义 S/M/C 程序的构建工作量。
4. 如果在过程数据库中没有类似的项目，并且不存在项目特性的基线(即，这个项目涉及到一个新领域或技术)，使用通用过程能力基线中 S/M/C 程序的平均构建工作量。
5. 使用项目特定的因素来进一步改进 S/M/C 程序的构建工作量。
6. 使用 S/M/C 程序的构建工作量和程序被调用的总次数来获得 总的构建工作量。

7. 使用能力基线或过程数据库中类似项目中的工作量分布数据来估计其他任务的工作量和总工作量。
8. 基于项目特定的因素重新改进估计。

#### 4. 自顶向下的工作量估计方法步骤（能够根据案例数据计算估计工作量）

1. 获得按照功能点计算的软件规模估计
2. 使用类似项目的 PCB 数据或 PDB 数据中生产率数据确定项目的生产率级别
3. 从生产率和规模估计获得整体的工作量估计
4. 使用从 PCB 获得的工作量分布数据估计各阶段的工作量
5. 修正估计，将项目特定因素考虑进去

#### 5. 掌握用 COCOMO 初级、中级模型估计工作量和开发时间（模型参数和查找表会给出，不需要记背）【单独的一个 PPT 里有解析】

- 某学院安装了一个工资系统，人事处要求创建一个子系统来分析每门课程的人力资源成本。要求该子系统提供查询每门课程人力资源成本的功能。每名教师所得工资的细节可以通过工资系统中的文件得到，教师花在教每门课上的小时数可通过一个基于计算机的计时表系统中的文件得到。该子系统将计算结果存放到由总会计系统读取的一个文件中，并产生一个报告，来显示每名教师每门课的课时数及这些课时数相应的成本。
- ①计算该子系统的 UFP、FP(子系统产生的报告复杂度为高，其它所有元素的复杂度均为中等)，开发语言自选。
- ②使用基本 COCOMO 模型估计工作量和进度。

**基本 COCOMO 模型**  $E = a \cdot S^b$   $D = c \cdot E^d$

E 表示工作量,单位是人月(PM)。D 表示开发时间,单位是月(M)。

S 是项目的代码行估计值，单位是千行代码 a,b,c,d 是常数

**中级 COCOMO 模型**  $E = a \cdot S^b \cdot EAF$   $D = c \cdot E^d$

工作量调节因子(EAF)

**软件产品属性：**软件可靠性、数据库规模、处理复杂度。

**计算机属性：**CPU 时间属性、内存容量约束、虚拟机发散性、回复时间的需求

**人员属性：**分析能力、应用经验、编程能力、虚拟机的经验、编程语言经验

**项目属性：**使用现代编程方法、使用软件工具、进度约束

$EAF = F1 \cdot F2 \cdot \dots \cdot F15$

正常情况下：Fi=1。Boehm 推荐的 Fi 值范围(0.70, 0.85, 1.00, 1.15, 1.30, 1.65)

#### 6. 制定项目时间进度计划进行的工作步骤

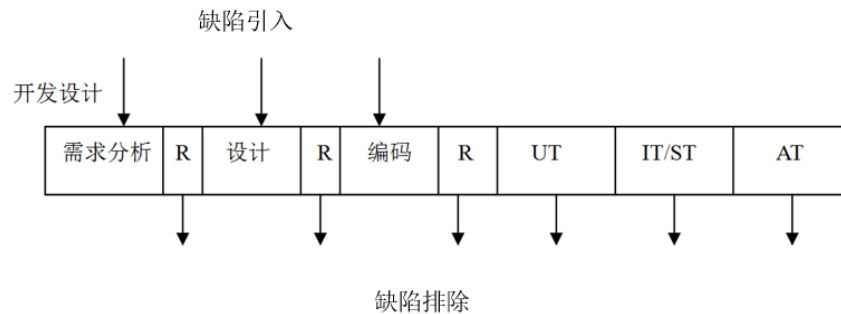
1. **定义活动：**列出为完成项目的各项须交付任务需执行的各具体项目活动(可跟 WBS 分解结合进行)
2. **排序活动：**确定各项目活动间的相互关系
3. **估算活动资源：**估算一个项目团队应该使用多少资源(人、设备和原料)来执行项目活动
4. **估算活动工期：**估算要完成各项目活动所需的时间
5. **制定进度计划：**分析活动排序、活动工期和资源需求，创建项目进度表
6. **控制进度：**控制和管理项目进度的变更。包括进度基准(更新)、请求变更、绩效衡量、推荐的纠正措施。

## 第七章：质量计划和缺陷估计

### 1. 软件质量的定义

**软件质量：**我们用已交付软件的缺陷(故障)密度作为软件质量的定义，即已交付软件中每个单位规模的缺陷(故障)数。

### 2. 缺陷注入和清除的环节



R-清除 UT-单元测试 IT-集成测试 ST-系统测试 AT-验收测试

### 3. 质量管理的主要任务

质量管理的任务是规划合理的质量控制任务，然后正确地执行和控制它们，以实现项目的质量目标。

### 4. 质量管理的程序化方法指什么，有什么不足

**程序化方法：**在定义好的点上执行一些过程以发现缺陷

不足：

- 缺少清除效率的量化描述
- 依赖规程的质量和规程执行的质量

### 5. 如何制定量化质量管理计划（能够根据案例数据估计质量目标）

- 设定质量目标 直接设定质量目标的方法
- 估计规模（代码行/功能点/生产率）
- 估计AT的缺陷数
- 各阶段的缺陷估计（PDB/PCB缺陷分布率）
- 质量过程计划
- 设定缺陷注入率（注入率= $D/S$ ）
- 估计规模S（代码行/功能点/生产率）
- 估计总缺陷数D
- AT的缺陷数D2（PDB/PCB缺陷分布率）
- 质量过程计划 直接设定缺陷注入率的方法

① 使用类似项目的以往数据进行 AT 缺陷估计

② 使用 PCB 数据进行 AT 缺陷估计

A. 质量目标为每功能点提交的缺陷数

1.按照每功能点(从 PCB)的缺陷数设定质量目标（每功能单元的缺陷数）

2.估计项目预期的生产率级别

3.按照(预期的生产率 x 预期的工作量)估计出按 FP 计算的规模

4.按照(质量目标 x 估计的规模)计算 AT 缺陷的数目

- B. 质量目标为缺陷清除率，结合缺陷注入率和估计规模进行估计
- 1.按照缺陷清除率设定质量目标（缺陷清除率=AT 前排除的缺陷/总缺陷数）
  - 2.估计项目预期的生产率级别
  - 3.按照(预期的生产率 x 预期的工作量)估计出按 FP 计算的规模
  - 4.估计总的缺陷数(缺陷注入率 x 估计的规模)
  - 5.根据缺陷总数和质量目标估计 AT 缺陷数

## 第八章：风险管理

### 1. 风险的含义

**风险：**风险是那些可能发生的事件或者条件，如果它确实发生了，则它的发生会对项目产生有害的或者负面的影响。另一方面，风险是一种概率事件，可能发生也可能不发生。

### 2. 风险管理的含义、内容、目标和特点【少内容】

**风险管理：**风险管理试图使由于意外事件而导致项目失败的概率降到最小。

**内容：**风险评估、风险控制（不是控制风险的发生而是控制风险的危害）

**风险管理的目标：**旨在识别出风险，然后采取措施使它们对项目的影响最小

**特点：**风险管理是要付出额外的成本；风险管理的价值不容易度量

### 3. 风险管理主要包含哪两类活动

风险评估和风险控制

### 4. 如何进行风险评估

**风险识别：**可能的风险检查表会谈、会议和头脑风暴计划、过程和工作产品的评审、调查

**风险等级划分：**

1. 对于每个风险，将风险发生的可能性划分为低、中或高这三个级别。如果需要，在每个给定级别的范围内分配可能性数值。
2. 对于每个风险，将它们对项目的影响评估为低、中、高或非常高这四个级别。如果需要。为后果的级别 1 到 10 分别分配一个权重。
3. 基于风险的可能性及其带来的后果划分级别---即，高可能性、高风险的条目会比中可能性和高风险的条目具有更高的级别。在发生冲突的情况下，使用判断（或用分配来计算风险代价值的数字）来决定。
4. 选取高层的几个风险来进行克服和跟踪。

### 5. 了解常见风险及其缓和措施

顺序号	风险类型	风险缓解步骤
1	人员流动的风险	确保关键项目领域有多个资源 有团队建设活动 在组员中进行工作轮换 为项目保留额外资源作为备份 维护个人工作的正确文档 严格遵守配置管理过程和指南
2	太多的需求变更	从客户那里获得对最初需求规格的签署 说服客户相信需求变更会影响进度 定义一个规程来处理需求变更 协商按照实际工作量付费
3	不清楚的需求	使用经验和逻辑制定一些前提假设并通知客户，获得签署开发原型或让客户对需求进行评审
4	缺乏有经验的技术人才	估计一些初始学习的容限 维护额外资源的余量 定义项目特定的培训程序 召开相互培训会议
5	强加于项目的外部决策	用所支持的事实或者数据列出本决策可能带来的风险，并与负责强加决策的人进行协商 如果是不可避免的，确定实际风险并给出相应的缓解计划

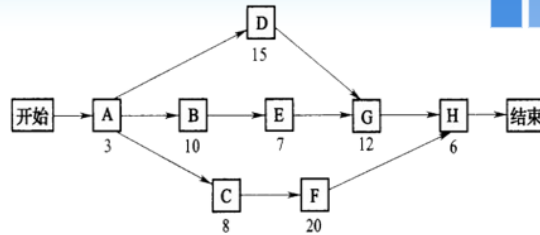
顺序号	风险类型	风险缓解步骤
6	不能满足性能需求	清晰定义性能准则并交给客户评审 定义必须遵守的标准来满足性能准则 准备满足性能准则的设计，并对其进行评审 模拟关键处理行为的性能或为其建立原型 在可能的位置，用有代表性的大量数据进行测试 在可能的位置执行压力测试
7	不实际的进度	商讨一个更好的进度 确定并行任务 及早准备资源 确定可以自动完成的领域 如果关键路径不在进度内，要与客户进行协商 协商按照实际工作量付费
8	使用新技术	考虑阶段交付 从交付关键模块开始 将学习曲线的时间包括在进度内 开发验证概念的应用
9	不充分的业务知识	增加与客户的交流并确保传递足够的知识 机构域的知识培训 模拟客户业务事物或为其建立原型，并获得批准
10	链路中段/性能下降	与客户一起设定正确的期望值 提前为链路负载进行计划 为优化使用链路进行计划

## 6. 关键路径计算方法

### 补充材料：关键路径

某项目的网络图如图所示，如果该项目的规定完工时间为42天，请确定关键路径。

如果时差都是正，则选择正时差值最小的活动；  
如果存在负时差，则选择负时差绝对值最大的活动。



时差计算表

活动	活动工期 DU	最早		最迟		总时差
		开始时间	完成时间	开始时间	完成时间	
		ES	EF	LS	LF	
A	3	0	3	4	7	4
B	10	3	13	7	17	4
C	8	3	11	8	16	5
D	15	3	18	9	24	6
E	7	13	20	17	24	4
F	20	11	31	16	36	5
G	12	20	32	24	36	4
H	6	32	38	36	42	4

## 第九章：项目管理计划

### 1. 项目管理计划的内容和使用者

**项目管理计划（PMP）：**是项目经理承担的所有规划任务的核心。各规划任务的结果都出现在 PMP 中，是指导所有项目执行的基准文档。

项目管理计划分四部分：

- 项目概述
- 项目计划
- 项目跟踪
- 团队

项目管理计划的主要使用者：

- 业务主管
- 项目经理
- 项目的开发人员

### 2. PMP 中的项目计划部分包含哪些内容

- 项目过程
- 工作量估计
- 开发环境
- 工具
- 培训计划
- 质量计划
- 里程碑
- 风险管理计划

### 3. PMP 中的项目跟踪包含哪些内容

- 任务跟踪
- 事宜跟踪



- 客户反馈
- 状态报告
- 升级规程

#### 4. PMP 中的项目团队包含哪些内容

- 项目机构
- 项目组
- 角色和职责

#### 5. 本章涵盖前述若干章节内容，要求能够进行综合分析【见 ppt 266 案例】

## 第十章：配置管理

### 1. 配置管理的概念

**软件配置管理（SCM）：**是项目管理的一项内容，主要涉及对变更进行系统地控制，建立和维护在项目的整个软件生存周期中软件项目产品的完整性。

主要包括：

- 标识在给定时间点上软件的配置
- 系统地控制对配置项的更改
- 维护在整个软件生存周期中配置的完整性和可跟踪性

### 2. 配置管理的功能

1. 给出程序的状态；
2. 给出一个程序的最新版本；
3. 处理并发更新申请；
4. 取消一个程序变更；
5. 防止未授权的变更或者删除；
6. 提供需求变更申请和程序变更之间的可跟踪性；
7. 取消一个需求变更；
8. 显示相关的变更；
9. 收集当前系统的所有源代码、文档和其他信息。

### 3. 典型的配置项有哪些

需求规格、设计文档、源代码、测试计划、测试脚本、测试规程、测试数据、项目使用的标准、验收计划、CM 计划、项目计划、用户手册、培训材料文档、合同文档、质量记录、CM 记录、已包含的软件产品。

### 4. 如何进行配置管理计划

1. 标识出典型的配置项
2. 配置管理员或者项目经理进行 SCM 计划
3. 配置管理计划阶段的任务：
  - 识别配置项，包括客户提供和购买的项
  - 定义为配置项命名和编号的计划
  - 定义 SCM 所需要的目录结构
  - 定义访问控制
  - 定义变更控制规程
  - 确定和定义 CC/CCB 的责任和权利
  - 定义跟踪配置项状态的方法

- 定义备份规程
- 如果需要，定义协调规程
- 定义发布规程
- 定义归档规程
- 确定将配置项转移到基线的点

## 5. 如何执行配置管理过程中的变更申请

1. 接受变更申请（影响分析之后）；
2. 建立一种跟踪机制；
3. 检出需要进行变更的配置项；
4. 执行变更；
5. 检入(注册)配置项；
6. 在项目的整个生命期内维护该配置项。

# 第十一章：评审

## 1. 评审的功能和特点

1. 评审可以应用于软件开发各个阶段、产生的各种类型产品——范围广；
2. 评审比软件测试更有效率，因为其看到的是问题本身而不是征兆；
3. 通过评审不仅可能发现错误，还可以提出对软件产品的改进意见，防止再发生；
4. 评审可以在产品开发阶段进行，作者对产品细节很清楚，可以及时修改；
5. 不只发现错误，还有利于评审员、软件项目相关组熟悉有关产品。

## 2. 了解基本的评审过程

小组评审的几个阶段：

- 评审规划
- 准备和概述
- 小组会议
- 返工及后续修改

## 3. 单人评审与组评审之间的区别

单人评审	评审能力基线一般包括什么
是正式评审 针对简单工作产品 1 位评审专家，提前预评材料 评审会议仅包含作者和评审专家 2 人	平均准备速度、 组评审会议的覆盖速度 缺陷密度(次要或普通缺陷、严重和主要缺陷) 整体缺陷密度

## 4. 评审能力基线一般包括什么

- 平均准备速度
- 组评审覆盖速度
- 不同级别的缺陷密度
- 整体缺陷密度

## 5. 评审分析指南（评审出的缺陷多于正常和少于正常的原因及措施，能够根据案例描述使用评审分析指南）

可能的原因	考虑采取的措施
<b>如果发现的缺陷少于正常情况</b>	
工作产品非常简单	将类似工作产品的组评审转变成单人评审（参照裁剪指南） 合并评审（参照裁剪指南）
也许没有仔细评审	检查覆盖速度；如果很低，重新安排评审，也许需要另外一支队伍
评审专家没有接受足够的组评审培训，或对评审的材料没有足够多的经验	安排或执行组评审培训 安排其他队伍进行重新评审
工作产品有非常好的质量	通过覆盖速度、作者、评审专家的经验等来确认这个事实；看该质量是否可以在项目的其他部分重现 修改随后活动的缺陷预测；看是否有通用的过程改进的经验
<b>如果发现的缺陷多于正常情况</b>	
产品质量很低	检查作者是否需要培训 重做产品 考虑重新分配将来的任务（即，只给作者安排更简单的工作）
产品非常复杂	确保在后续阶段有良好的评审或测试（参考裁剪指南） 为系统测试增加估计 将产品分成更小的组件
有太多的小错误（以及很少的大错误）	确定小错误的原因；通过适当增强检查表和使作者意识到常见错误原因，使错误在以后得到改正 评审者也许对工作产品没有充分的理解。如果如此，召开一个简短会议或用不同的评审专家另外进行评审
评审时所使用的参考文档不够详细和清楚	使参考文档得到评审并获批准
评审过的模块是项目中的第一批模块	分析缺陷、更新评审检查表并通知开发者 安排培训

## 第十二章：项目监督和控制

### 1. 项目监督和控制的主要活动

数据采集、项目跟踪、量化监督和控制（里程碑分析）、缺陷分析和预防

### 2. 数据采集的内容

1. **工作量数据**：通过日报和周报采集（活动编码），评价项目是否在预算内执行。
2. **缺陷数据**：缺陷数据库，缺陷类型定义，缺陷级别定义
3. **规模测量**：功能点、代码行

### 3. 项目跟踪的内容

- 任务跟踪（MSP）
- 缺陷(故障)跟踪（DCS）
- 问题跟踪（事宜跟踪）
- 状态报告

### 4. 里程碑分析指南（工作量/进度、测试性能，能够根据案例描述使用里程碑分析指南）

表 13.7 工作量/进度性能指南

可能的原因	考虑采取的行动
如果实际的比估计的少的程度超过了允许的限度	
程序的估计太高, 或者项目组在该领域内的知识或经验比预想的更多	为以后的模块重新估计 释放资源
迄今为止没有充分地执行当前的任务	评审迄今为止已完成的任务, 并为没有评审的工作产品安排评审 检查事宜记录
如果实际的比估计的多的程度超过了允许的限度	
该领域内知识匮乏	安排培训
作者的软件/编码经验不足	重新分配以利用现有经验
新技术领域	重新估计或请求资源, 协商交付日期
估计太紧凑了	确定主要组件的额外工作量并为将来的活动更改估计 请求资源 协商降低项目目标
资源优化差	重新安排任务并重新确定任务的优先级, 确定和消除“等待时间”
不能获得关键资源	向上级反映该事宜 获得备份资源 重新安排进度, 谨记关键资源
一些早期阶段的输出质量很差, 引起太多的返工	确定问题的来源并纠正它们 更改项目进度

表 13.8 测试性能指南

可能的原因	考虑采取的行动
如果实际发现的缺陷数小于估计的数目	
高质量的工作产品	确定原因并查看对项目或过程有无可取的教训
不充分的测试	检查花费在测试上的工作量; 评审并加强测试计划 安排更多的测试
早期质量控制活动的执行非常细致	分析项目所有的评审和测试记录 检查项目在过程中是否有可取的教训
缺陷估计过高	确定原因, 并改正估计
如果实际发现的缺陷数大于估计的数目	
到目前为止, 质量控制活动的不适当执行, 或者不充分的评审和单元测试	检查所有测试和评审记录 在继续测试前安排关键模块的评审 加强测试计划和安排更进一步的测试 评审估计和为验收制定的计划
缺陷估计太低	确定原因, 并改进估计

## 5. 常见缺陷分析的方法

- 帕累多分析: 找到主要缺陷 (类)
- 因果分析: 找到主要原因

# 第十三章：项目收尾

## 1. 项目收尾的主要活动

1. 项目收尾费分析: 执行收尾分析+收尾分析报告
2. 归档:
  - 项目初始化: 建议书、合同文档、评审记录
  - 需求规格: 数据流图、实体关系图、数据字典、需求文档、分析文档、评审记录
  - 计划文档: 项目计划、配置管理计划、安装和验收计划、维护计划、评审记录
  - 概要设计: 数据库设计、功能设计、环境结构设计、测试用例设计和检查表、项目标准、评审记录
  - 程序规格: 流程图/结构表、伪代码、程序规格文档、评审记录

- **源代码：**源文件、副本、数据库定义、批处理文件、用于生成开发环境的文件（脚本文件、登录文件）、可执行程序
- **和测试相关的文档：**模拟程序、测试数据、测试用例、测试结果等
- **验收：**验收规程、验收记录
- **手册：**用户手册、操作指南、安装手册、命令参考手册
- **配置管理：**配置管理环境定义规程等

## 2. 参与执行首尾分析的角色

质量顾问、SEPG、项目经理、业务主管等

## 3. 收尾分析报告主要内容

与过程相关的通用数据、风险管理、规模、工作量、缺陷、原因分析、过程资产

# 考试形式

1. 闭卷考试，考试时长 2.5 小时
2. 总成绩=卷面成绩 80%+平时成绩 20%
3. 考试题型包括：填空题、判断题、名词解释、问答题、案例分析题、论述题
4. 考试时间为结课后 3 周左右，具体时间地点请关注教务老师的通知