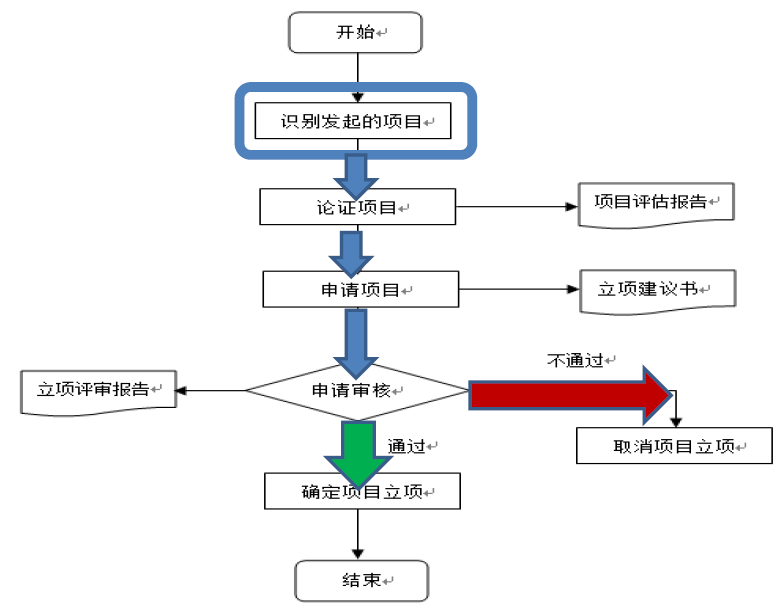
第一章

项目和日常工作的区别：项目有时限性和唯一性，项目管理以目标为导向，项目完成由项目经理以及项目团队，存在大量变更管理。

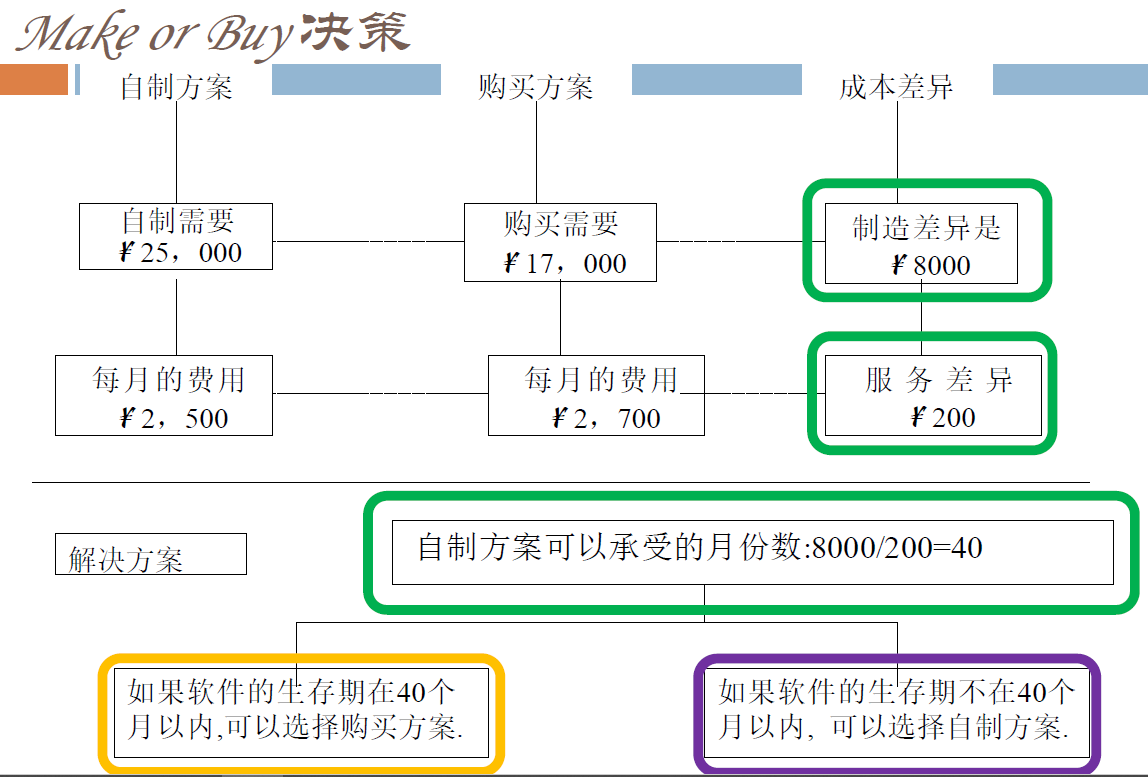
日常工作具有连续性、重复性，日常管理通过效率和有效性体现。大多是职能式的线性管理，基本保持持续性，连贯性。

第二章

项目的立项：明确项目的目标、时间表、项目使用的资源和经费，而且得到项目发起人和项目经理的认可。

流程：

Make跟buy的策略：如何选择



招标投标过程：我们来看一下招投标的过程。

首先甲方确定招标书，其中包括了甲方的需求，即需要采购的内容，软件项目的验收标准等等。接下来进行招标信息的发布，招标的方式有很多，比如公开招标，有限招标，多方洽谈和直接谈判等。潜在的乙方看到招标之后，进行项目分析，如果认为可以参加竞标，需要提交标书，即建议书。然后，甲方主持评标，选出合适的乙方，签署合同。

甲方招标书定义（说明招标者的需求，让参与投标的单位按照规范的格式准备投标书）→招标（招标者要向供应商发出正式的投邀请，并向他们发送（或发售）招标文件）→乙方项目分析与竞标（供应商在得到招标文件后，要仔细分析项目需求，并进行可行性研究，在做出项目可行的结论并决定参与投标后，就要组织人员编写投标书）→评标（根据招标文件中规定的条件对投标者进行评估，从中选择能够提供最优服务和最合理价格的供应商）→合同签署。

合同的两种类型：固定价格合同，成本加酬金合同。

项目的章程：确认项目存在的文件，包括对项目的确认对项目经理的授权和项目目标的概述等。

第三章

模型的特点和适用场景：

在预测模型中，项目的执行通常用顺序的方式执行，没有迭代。

(1)瀑布模型

适用于软件需求很明确的软件项目，即一般适用于功能明确、完成、无重大变化的软件系统的开发，即：

1) 在项目开始前，项目的需求已经被很好的理解、也很明确，而且项目经理很熟悉为实现这一模型所需要的过程。

2) 解决方案在项目开始前也很明确。

3) 短期项目可采用瀑布模型。

严格按照单向的顺序执行。

优点：管理方便

缺点：不易改变

1. V模型

是瀑布模型的变种，强调 开发与测试的伴随关系，

适用于项目需求在项目开始前很明确、解决方案在项目开始前也很明确，项目对系统的安全很严格，如航天飞机控制系统、公司的财务系统等。

1. 迭代模型通过连续的原型和概念验证来改进产品或者结果。

迭代模型不断对部分完成或未完成的成果进行反馈，从而对该工作进行改进和修改。

有助于减少项目的不确定性。也称为原型模型。

1. 增量模型就是一个增量一个增量的开发过程，每个增量是一个交付成果。

每一个增量都包括：分析、设计、实施、测试、提交等过程。

优点是：加快交付速度，频繁的交付少量的成果，可以避免一次性投入太大带来的风险

1. 敏捷模型(Agile Development)是一种以人为核心、迭代的开发方法。

第一点，就是尽早，持续提交有价值的成果。

第二，提出要适应需求的变更。

第三个原则，强调短周期，经常性地提交可运行的产品。

第四个原则，强调客户参与开发工作。

第四章

需求分析的过程：需求获取，需求分析，规格说明，需求验证

需求分析工作完成的一个基本标志: 是形成了一份完整的、规范的需求规格说明书。

变更控制委员会(SCCB) 1评估变更2批准变更申请3在生存期内规范变更申请流程4对变更进行反馈5与项目管理层沟通

第五章

工作分解结构法，WBS，把一个项目，按一定的原则分解，项目分解成任务，任务再分解成一项项工作，再把一项项工作分配到每个人的日常活动中，直到分解不下去为止。即:项目→任务→工作→活动。

工作包是分解结果的最小单元，是WBS的最低层次可交付成果。工作包应当由唯一主体负责。

WBS有哪些方法：

类比：有些项目有相同或相似的周期，因此而形成相同或相似的工作细目。这些项目进行任务分解的时候，可以采用类比方法。  
模板参照：第二个分解方法是模板参照方法，如果某个项目有可以参照的wbs模板，在任务分解时就可以采用 模板参照的方法，比较方便。

自上而下：是对项目有大体把握、了解和熟悉的。是最主要和常见的任务分解方法，从项目的大局着手，按照一定的逻辑和结构分解子项目。

自下而上：（对项目需求不太清晰）

第六章

软件项目规模度量单位：LOC ：源代码程序长度的测量.。FP：系统功能数量的测量。

人月、人天、人年:人在单位时间内完成的任务量

工作量和成本的关系：工作量是成本的主要考虑因素，项目的工作量估算和成本估算常常同时进行。如果确定了单位工作量的成本，则可根据项目工作量直接计算出项目成本。

项目成本的构成：直接成本:与具体项目的开发直接相关的成本。如人员的工资、外包外购成本等。又可细分为开发成本、管理成本、质量成本等。间接成本:不归属于一个具体的项目，是企业的运营成本，分摊到各个项目中。如房租、水电、保安、税收、福利、培训等等。

项目成本=直接成本＋间接成本

直接成本=开发成本＋管理成本＋质量成本。

间接成本=直接成本\*间接成本系数。

工作量估算的方法：代码行、功能点。类比（自上而下）估算法参数估算法。专家估算法

代码行。优点：直观、准确（在有代码的情况下）、易于计算（可使用代码行统计工具）。缺点：对代码行度量没有公认的标准定义。代码行数量依赖于所用的编程语言和个人的编程风格。在项目早期,需求不稳定、设计不成熟、实现不确定的情况下很难准确地估算代码量。

Fp计算题

COCOMO模型按详细程度分：基本COCOMO模型是一个静态单变量模型，它用一个已估算出来的原代码行数为自变量的经验函数计算软件开发工作量。中级COCOMO模型在基本COCOMO模型的基础上，再用设计产品、硬件、人员、项目等方面的影响因素调整工作量的估算。详细COCOMO模型包括中间COCOMO模型的所有特性，但更进一步考虑了软件工程中每步骤(如分析、设计）的影响。

按项目类型分：组织型(organic) :相对较小、较简单的软件项目、开发人员对开发目标理解比较充分，与软件系统相关的工作经验丰富，对软件的使用环境很熟悉，受硬件的约束较小，程序的规模不是很大(<500o行)。嵌入型(embedded):要求在紧密联系的硬件、软件和操作的限制条件下运行，通常与某种复杂的硬件设备紧密结合在一起。对接口、数据结构、算法的要求很高、软件规模任意。如大型/超大型操作系统，航天用控制系统等。半独立型(semidetached):介于上述两种软件之间。规模和复杂度都属于中等或更高。最大可达30万行。

专家估算考计算。

预算估算的不同：成本估算的目的是估计项目的总成本和误差范围，而成本预算是将项目的总成本分配到各工作项上。成本估算的输出结果是成本预算的基础与依据，成本预算则是将已批准的估算进行分摊。尽管成本估算与成本预算的目的和任务不同，但两者都以工作分解结构为依据，所运用的工具与方法相同，两者均是项目成本管理中不可或缺的组成部分。

成本基线是成本预算的成果之一，是项目从开始到结束的整个生命周期内的成本累计曲线，它描述了项目生命周期中到某个时点为止的累计成本支出。原始的成本预算就是成本基线，也就是项目的期望成本。

第七章

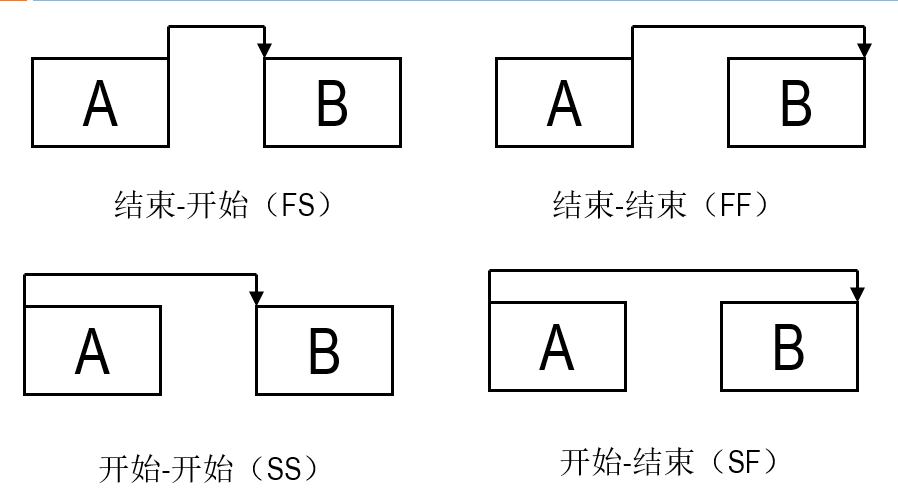
活动间的关系：

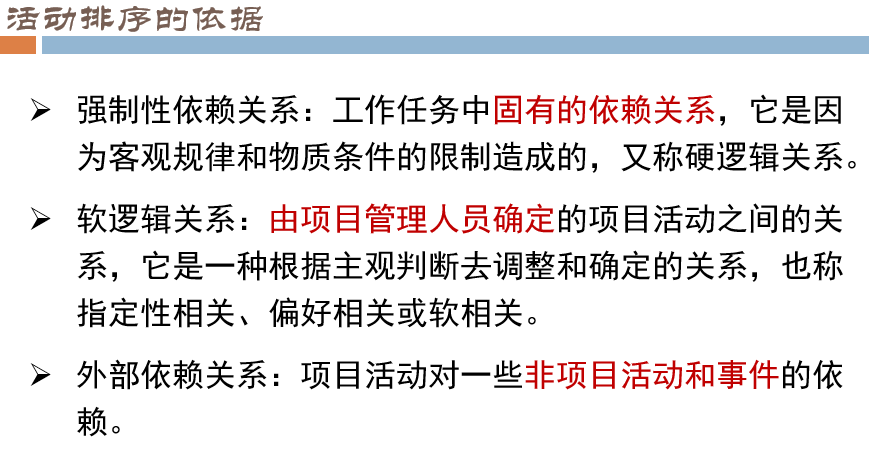
结束-开始：A活动结束的时候，B活动才能开始。这是最常见的逻辑关系。

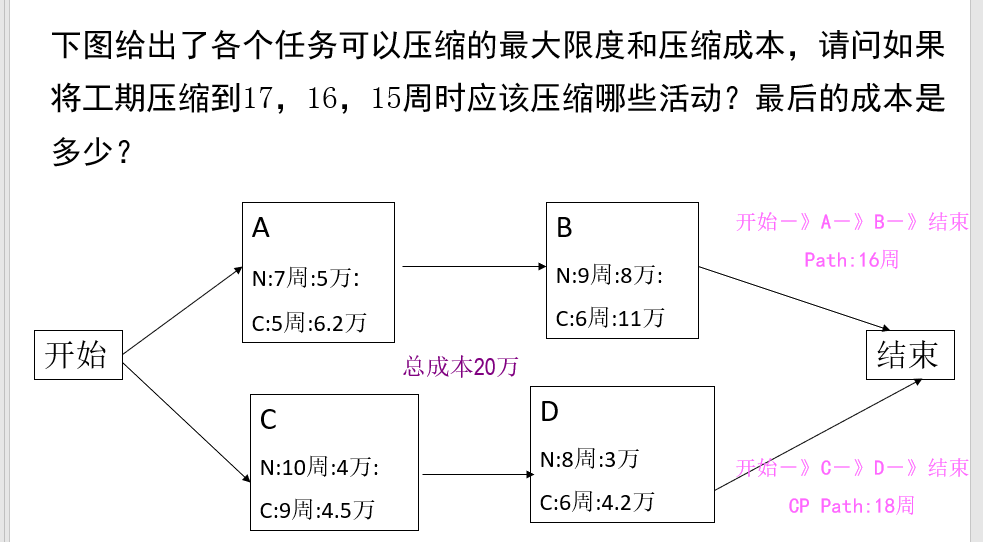
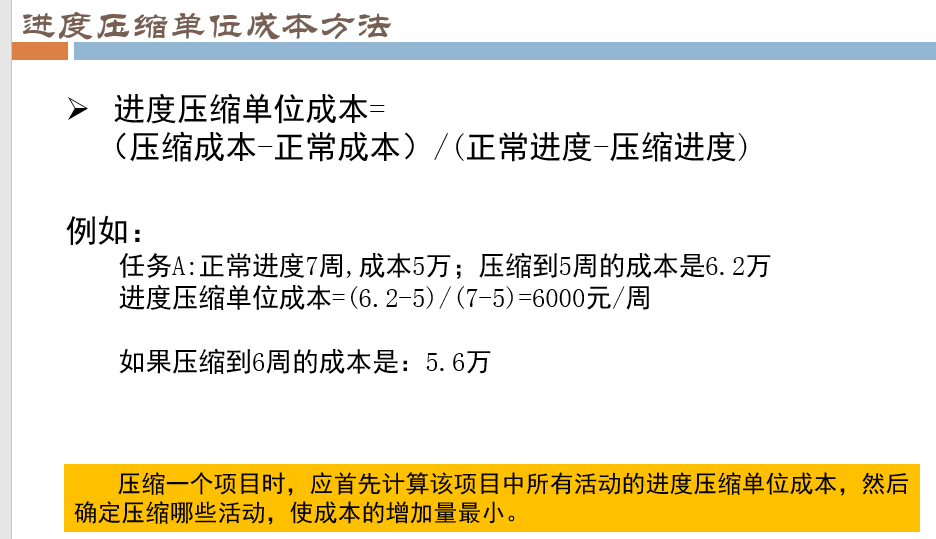
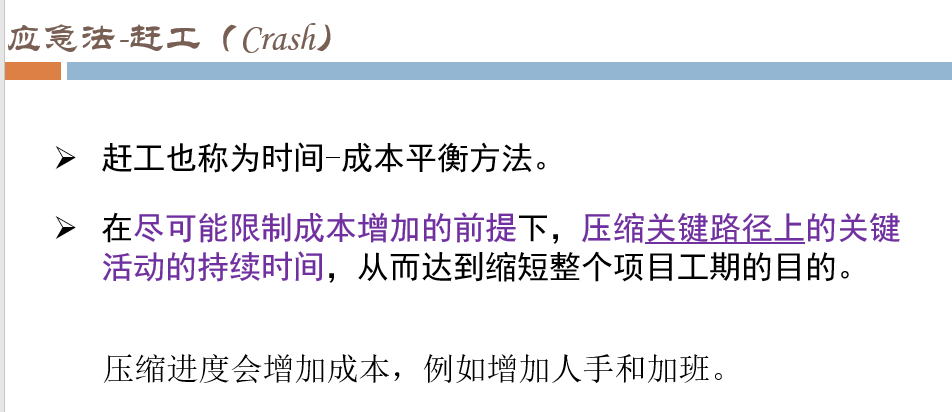
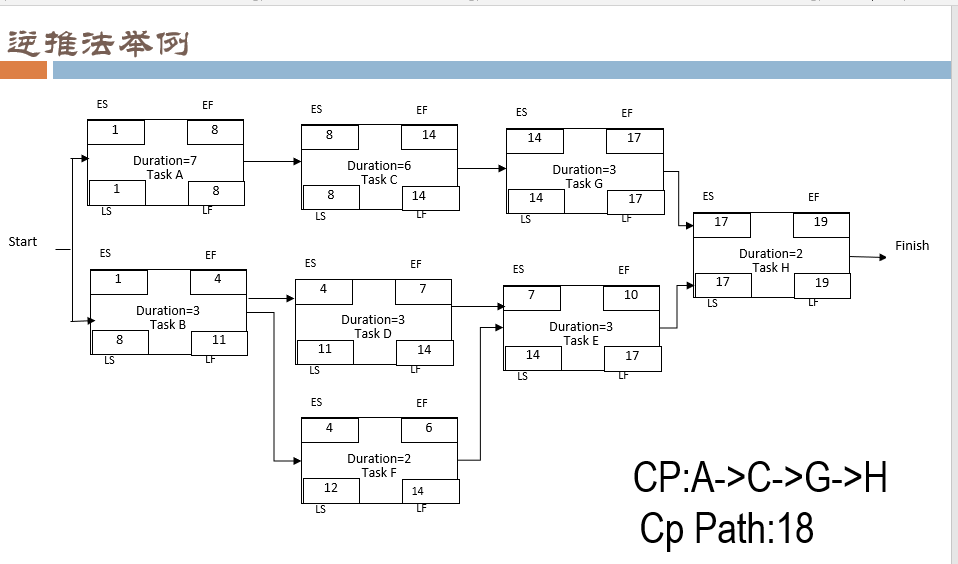
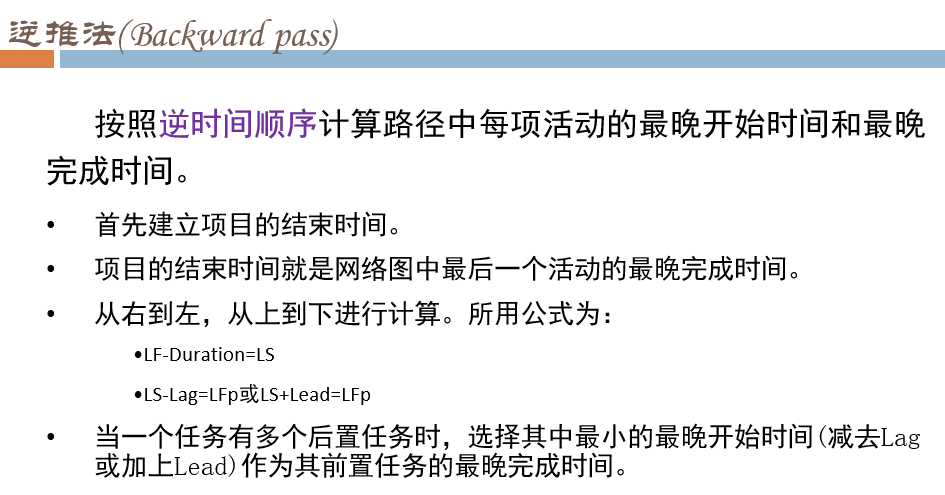
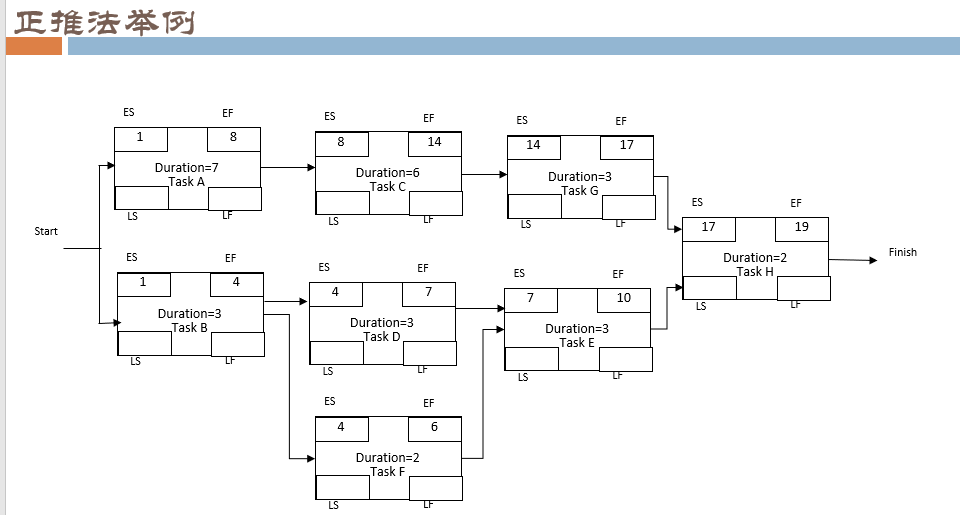
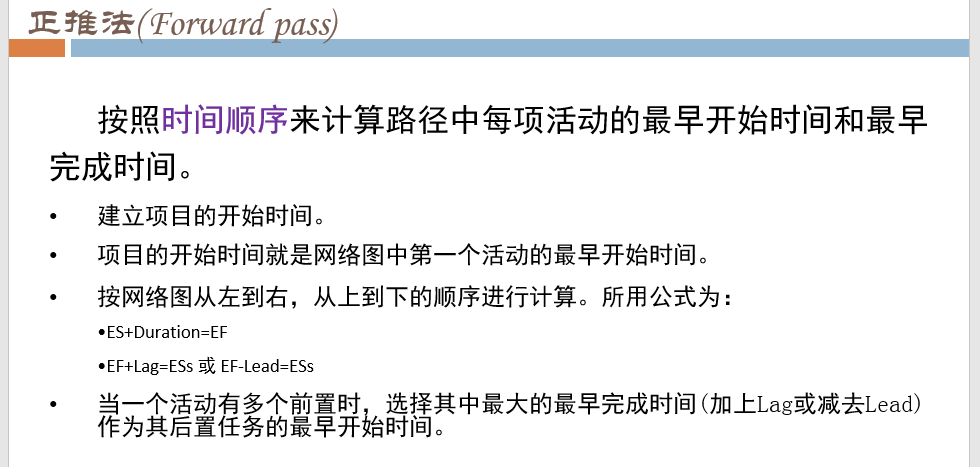
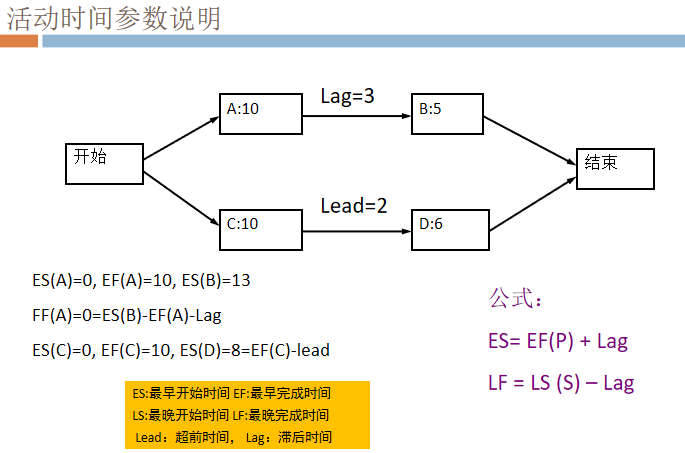
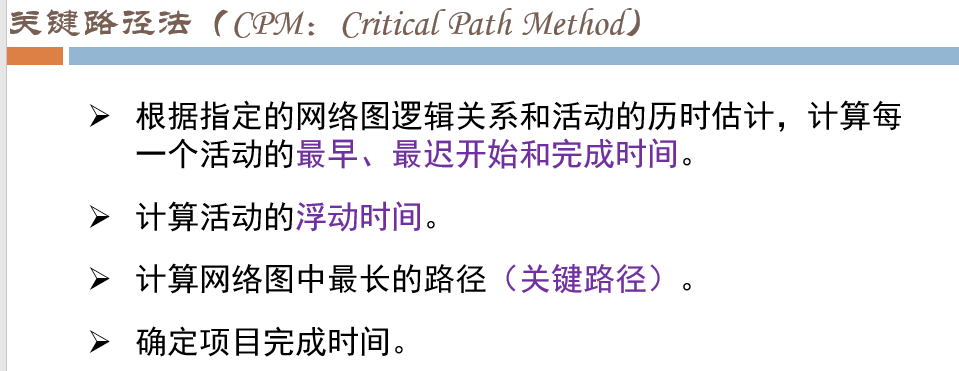
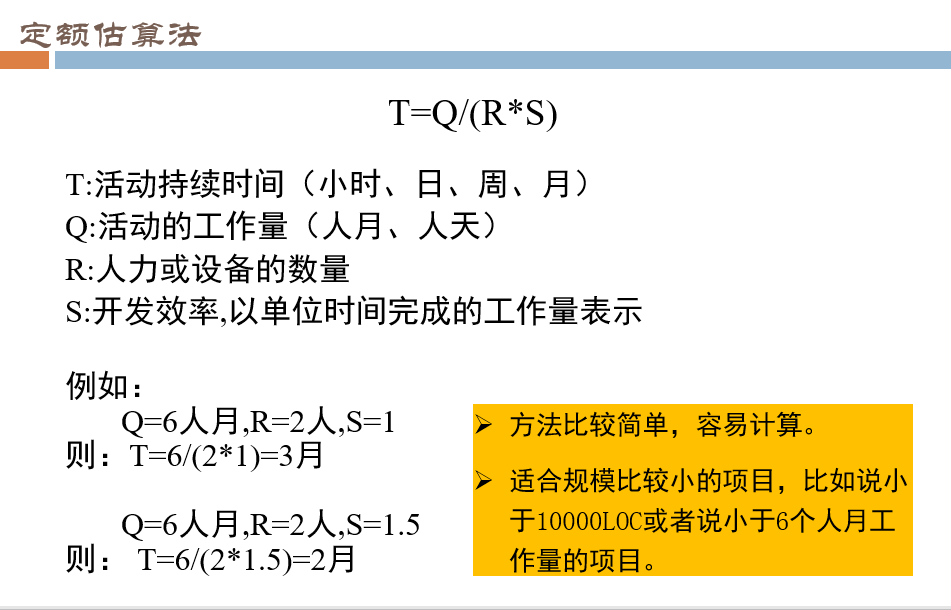
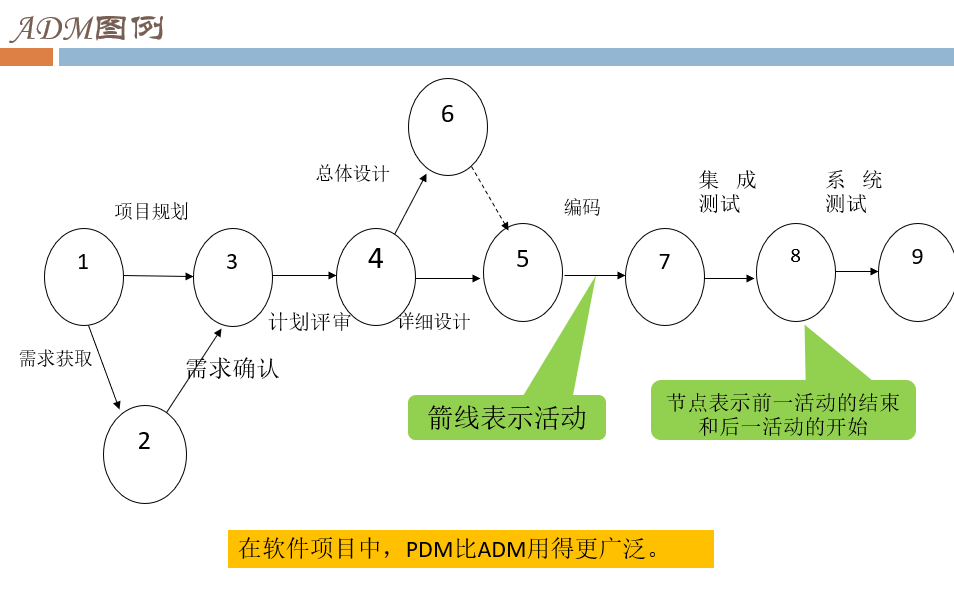
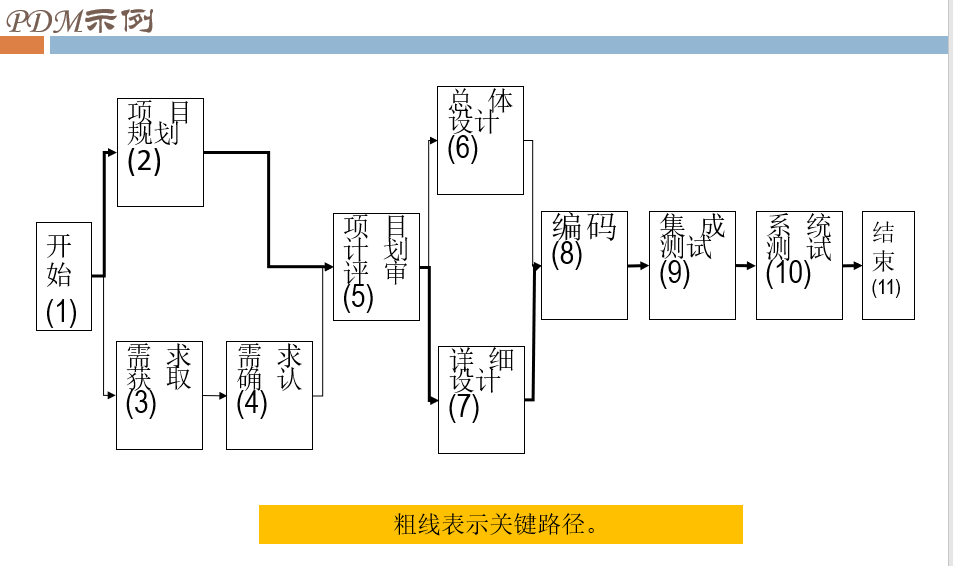
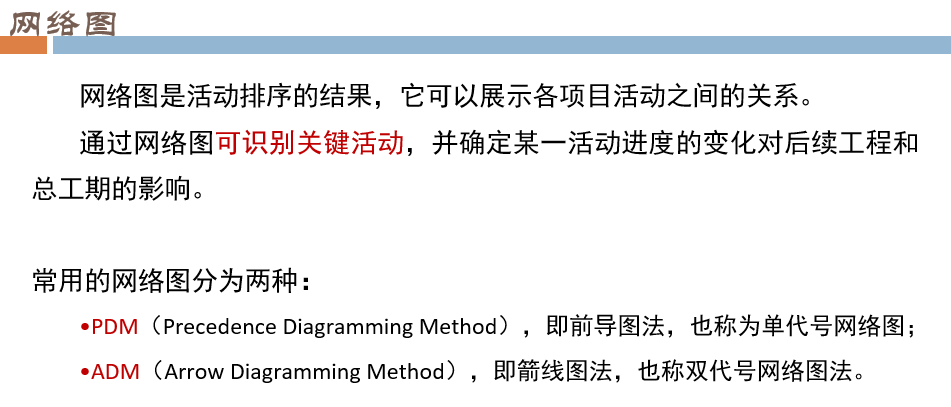
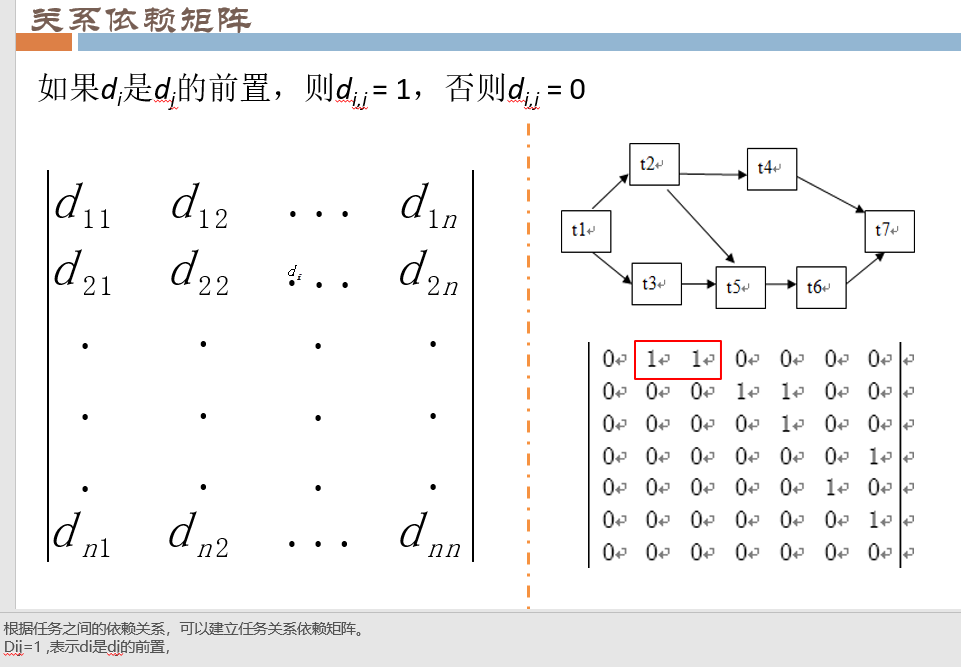
结束-结束：B活动的结束必须等到A活动的结束。

开始-开始：A活动开始的时候，B活动也开始。

开始-结束：A活动开始的时候，B活动结束。极少出现这种关系。







第八章

软件质量是软件满足明确说明或者隐含的需求的程度。

审计：将审核的主体与为该主体以前建立的一组规程和标准进行比较，已确保遵循这一规程和标准。

质量保证（QA）：是对完成工作的检查，这个任务本身并不能提高产品的质量,提出一些不合格项，做为下一次版本的改进，一般由质量保证部门人员实施。

质量控制（QC）：通过识别和纠正开发软件中的缺陷来确保软件质量，由测试团队负责。测试关心的不是过程的活动，而是对过程的产物以及开发出的软件进行剖析，走查，运行，以找出问题，报告质量。技术评审，代码走查，测试

第九章

配置基线：基线提供了软件生存期中各个开发阶段的一个特定点,标志开发过程一个阶段的结束,或者里程碑，一个(些)配置项形成并通过审核，即形成基线。基线修改需要按照正式的程序执行

第十章

团队计划几种形式：

职能型：

优点：

1以职能部门作为承担项目任务的主体，可以充分发挥职能部门的资源集中优势，有利于保障项目需要资源的供给和项目可交付成果的质量。

2职能部门内部的技术专家可以同时被该部门承担的不同项目所使用，节约人力，减少了资源的浪费。

3同一职能部门内部的专业人员便于相互交流、相互支援，对创造性地解决技术问题很有帮助。

4当有项目成员调离项目或离开公司，所属职能部门可以增派人员，保持项目的技术连续性。5项目成员可以将完成项目和完成本部门的职能工作融为一体，可以减少因项目的临时性给项目成员带来的不确定性。

缺点：

1客户利益和职能部门的利益可能发生冲突，职能部门可能会为本部门的利益而忽视客户的需求。

2当项目需要多个职能部门共同完成时，或者一个职能部门内部有多个项目需要完成时，资源的调配就会出现问题。

3当项目需要由多个部门共同完成时，权力分割不利于各职能部门之间的沟通交流、团结协作。

4项目成员在行政上仍隶属于各职能部门的领导，项目经理对项目成员没有完全的领导权。

项目型：

优点：

1项目经理对项目可以全权负责，可以根据项目需要随意调动项目组织的内部资源或者外部资源。

2项目型组织的目标单一，完全以项目为中心安排工作，能够对客户的要求做出及时响应，有利于项目的顺利完成。

3项目经理对项目成员有完全的领导权，项目成员只对项目经理负责，避免了职能型项目组织结构下项目成员处于多重领导、无所适从的局面，项目经理是项目的真正、唯─的领导者。4组织结构简单，项目成员直接属于同一个部门，彼此之间的沟通交流简洁、快速，提高了沟通效率，同时也加快了决策速度。

缺点：

1不同的项目组织，资源不能共享，即使某个项目的专用资源闲置，通常也无法应用于另一个同时进行的项目，人员、设施、设备可能会重复配置，造成一定程度的资源浪费。

2公司里各个独立的项目型组织处于相对封闭的环境之中，公司的宏观政策、方针很难做到完全、真正的贯彻实施，可能会影响公司的长远发展。

3在项目完成以后，项目型组织的使命即完成，项目成员有可能被解雇，对项目成员来说，缺乏—种事业上的连续性和安全感。

4公司承担的项目之间处于一种条块分割状态，项目之间缺乏信息交流。

矩阵型：

优点：

1专职的项目经理负责整个项目，以项目为中心。公司的多个项目可以共享各个职能部门的资源。

2专职的项目经理负责整个项目，以项目为中心。公司的多个项目可以共享各个职能部门的资源。

3既有利于项目目标的实现，也有利于公司目标方针的贯彻。

4项目成员在事业稳定性上的顾虑减少了。

缺点：

1容易引起职能经理和项目经理权力的冲突。

2容易引起职能经理和项目经理权力的冲突。资源共享也能引起项目之间的冲突。

3项目成员有多头领导。



结束

收尾：收尾过程是项目干系人和客户对最终产品进行验收，使项目有序地结束的过程。

移交：在项目收尾阶段，如果项目达到预期的目标，就要进行移交过程。

清算：项目清算是非正常的项目终止过程。