



北京交通大学
BEIJING JIAOTONG UNIVERSITY



软件系统分析与设计 System Analysis & Design

M210007B [03]

Jingxin Su & Haiming Liu

Sunday, May 29, 2022

/-1 软件项目管理

Software Project Management

软件项目管理目标

通常认为，项目成功的标志，也是项目管理人员争取的目标，应该包括以下几个方面

- 1. 达到项目预期的软件产品功能和性能要求**
- 2. 时限要求，项目应在合同规定的期限内完成**
- 3. 项目开销限制在预算之内**

软件项目管理涉及的范围

软件项目管理涉及的几个主要方面是人员、产品、过程和项目，即所谓4P

1. 人员 (People)
2. 产品 (Product)
3. 过程 (Process)
4. 项目 (Project)

人员

美国卡内基·梅隆大学软件工程研究所的Bill Curtis在1994年发表了“人员管理能力成熟度模型” (people capability maturity model, P-CMM)。该模型力图通过吸引、培养、激励、部署和聘用高水平的人才来提升软件组织的软件开发能力

人员管理涉及：利益相关方；团队负责人；团队集体

利益相关方，包括：

1. 项目的高级管理者
2. 项目经理
3. 开发人员
4. 客户
5. 最终用户

人员

团队负责人，在小项目的情况下，项目经理就是团队负责人。而大型项目也许会有若干个设计、编程团队或是若干个测试团队。

团队集体，团队内部有分工是必要的，但必须很好地配合，做到步调一致，为此必须强调以下3点：

1. 个人的责任心，这是团队完成工作的基本条件
2. 互相信任、尊重以及互相支持
3. 充分的交流与沟通

产品

项目经理必须在项目开始时就明确项目的以下三个目标：

- 1. 产品的工作环境。**
- 2. 产品的功能和性能。**
- 3. 产品工作处理的是什么数据，经它处理后得到什么数据。**

只有明确了项目的这些基本要求才能着手项目管理的各项工作，如项目估算、风险分析、项目计划的制定等

过程

过程在软件工程项目中是重要的因素，它决定着项目中开展哪些活动以及对活动的要求和开展活动的顺序

项目

项目管理的任务是如何利用已有的资源，组织实施既定的项目，提交给用户适用的产品

项目管理要开展的主要工作可分为3类。

- 1. 计划及计划管理。包括项目策划及计划制定、项目估算、风险分析及风险管理、进度管理、计划跟踪与监督**
- 2. 资源管理。包括人员管理（人员安排、使用）、成本管理、信息管理**
- 3. 成果要求管理。包括需求管理、配置管理、质量管理**

项目估算

通常在项目的目标确定和软件基本功能确定之后，就应该着手项目计划的制定工作。

项目估算是制订计划的基础和依据

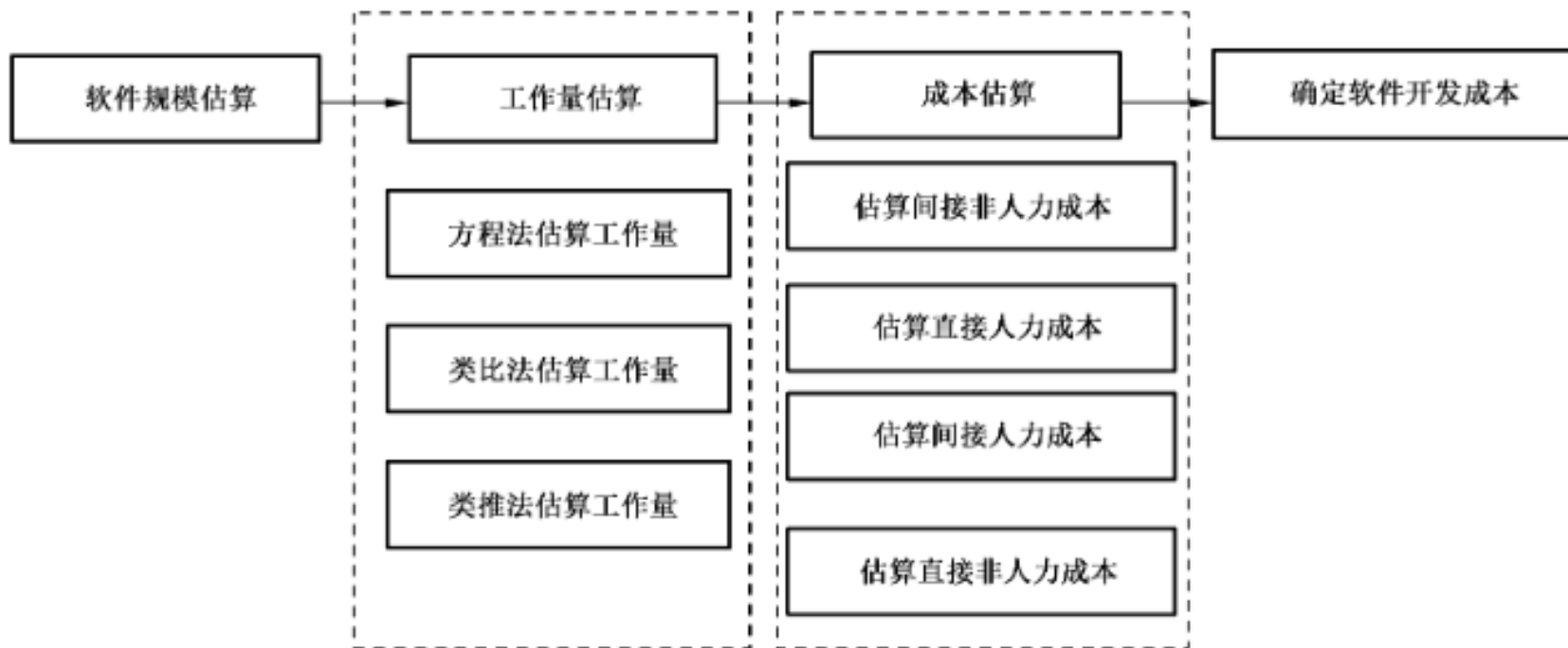
项目策划是项目开展初期阶段的重要工作，其主要目标是得到项目计划，或者说计划（plan）是策划（planning）的结果

项目策划中需要开展的活动

1. 确认并分析项目的特征
2. 选择项目将遵循的生存期模型，确定各阶段的任务
3. 确定应得到的阶段性工作产品以及最终的产品
4. 开展项目估算，包括估算产品规模、工作量、成本以及所需的关键计算机资源
5. 制订项目进度计划
6. 对项目风险进行分析
7. 制订项目计划

项目估算中的重要属性

在项目估算中，要解决的问题是项目实施的几个主要属性，即将要开发产品的规模（size）、项目所需的工作量（effort）以及项目的成本（cost）



项目估算中的几个重要属性

- 1. 规模：**项目的规模指的是得到最终软件产品的大小。一般以编程阶段完成以后得到程序的代码行表示，如以1千行代码为单位，记为KLOC。当然，在项目的开始只是对代码行的估计值。另一表示方法是功能点，记为FP（function point），它是根据软件需求中的功能估算的
- 2. 工作量：**项目的工作量按项目将要投入的人工来考虑，以一个人工作一个月为单位，记为“人月”
- 3. 成本：**软件项目的成本通常只考虑投入的人工成本，如某项目投入的总人工费用为12万元

功能点方法

功能点方法（function point）简称FP方法，该方法克服了项目开始时无法得知源程序行数的实际困难，从软件产品的功能度（functionality）出发估算出软件产品的规模

ICS 35.080
L 77
备案号：

SJ

中华人民共和国电子行业标准

SJ/T 11463—2013

软件研发成本度量规范

Specification for software research and development cost measurement

成本计算

一个软件组织在完成多个项目以后积累了一些数据，进行成本分析后可得到自己的生产率数值和人工价格

生产率是平均每个人月完成的源程序行数，可记为KLOC/人月或FP/人月

人工价格则为每人月的价值

有了这两个数值，如果在估出项目规模以后就可以很容易得到项目的工作量和成本，即

$$\text{工作量} = \text{规模} / \text{生产率}$$

$$\text{成本} = \text{工作量} \times \text{人工价}$$

风险管理

软件工程过程中可能出现的那些影响软件目标实现或是可能造成重大损失的事件称为软件风险

在软件开发项目的最初阶段，确认需求对整个开发工作是至关重要的。软件开发项目经常遇到的一个严重问题就是用户的需求一变再变。可以说这是一个典型的软件风险

软件工程项目所需的主要资源是合格的人员，有不少软件项目可能出现合格人员短缺的现象，这对于达到项目的目标自然构成威胁。

风险特点

在软件项目实施过程中会遇倒很多事件，软件事件和软件风险需要区分开，具有以下特点

- 1. 可能发生的事件**
- 2. 会给项目带来损失的事件**
- 3. 可能对其加以干预**

风险分类

依据危害性分类，从危害到软件项目本身讲，软件风险可分为3类：

- 1. 成本风险。成本风险是项目预算和开销不够准确造成的**
- 2. 绩效风险。绩效风险是系统不能提供全部或某些预期效益，或是不能实现预期的软件需求**
- 3. 进度风险。进度风险关系到项目进度或项目达到指定里程碑的不确定性**

风险分类

依据范围分类：

- 1. 项目风险。**这种风险涉及预算、成本、进度、人员的招聘和组织、资源的获取，以及顾客和需求等方面的问题
- 2. 技术风险。**技术风险威胁着开发产品的质量和交付产品的时间。技术风险会涉及设计方案、实现、接口、验证以及维护等方面的问题
- 3. 商业风险。**商业风险的发生会威胁开发软件的生命力，它会危及软件项目和产品出路。如市场风险、策略风险、管理风险、预算风险

风险管理的任务

风险管理的目标

1. 识别风险。识别风险是要找出可能的风险，对其进行分析、评估，并进一步对这些风险排序，以突出最为险恶的风险
2. 采取措施，把风险造成的影响降低到最小

风险管理的策略

1. 回避风险。例如改变项目的某些功能或性能需求使风险不可能发生
2. 转移风险。把风险转移到其他系统，或是借助购买保险将经济损失转移，从而化险为夷
3. 承受风险，接受风险，但将风险损失控制在项目资源可承受的范围之内

风险评估

风险评估的目标是认识可能的风险，它应该是风险控制的前提。

风险评估通常包括：

- 1. 风险识别**
- 2. 风险分析**
- 3. 风险排序**

风险识别

风险识别是风险评估的第一步。就某个特定的软件工程项目来说，从项目的具体情况出发，列举出可能出现的风险，真正弄清每一可能风险的情况是风险识别的主要任务。检查单(checklist)是识别风险的有力工具。采用检查单来识别风险是将检查单中所列举的各种风险，对照即将开发的软件项目，逐一加以甄别，判定检查单中哪些风险在该项目中可能发生。

在进行风险识别时采用访谈、调查还是会议的方式，或是对计划、过程和工作产品进行评审的方式，均应根据软件项目的具体情况决定。

风险分析

- 1. 风险分析的任务是分析每个风险可能造成的影响，给出风险大小的量值**
- 2. 进行分析可以借助一些已有的模型，但也并非所有已列出的风险都可借助模型进行分析，因此常常采用主观分析**
- 3. 风险分析方法包括：COCOMO成本模型，判定分析，网络分析，质量因子分析，性能分析等**

风险排序

- 1. 识别并分析风险将使我们初步弄清可能妨碍达到项目目标的危险事件，然而各种风险的后果会有很大的差别**
- 2. 我们必须对其加以区别，以便把管理者的目光集中到最高风险的事件上。这里所说的风险高低是以风险显露造成损失的大小来衡量，损失大的风险自然应该给予更多的重视**

风险控制

风险管理策划是要针对每个已经过识别和分析认为应该受控的风险制定风险管理计划

按Boehm的意见，风险管理计划主要包括以下5个方面：

- 1. 该项风险为什么重要，为什么一定要管理**
- 2. 风险管理应该能够提供什么以及什么时候提供**
- 3. 实施这些风险管理活动的责任人是谁**
- 4. 风险怎样能够得到减轻，该采取什么措施**
- 5. 需要什么资源**

风险化解

风险化解是要实际消除风险或减轻风险。实施风险管理计划从根本上讲就是将风险化解

为了帮助选择风险减轻的方法，必须考虑减轻风险的成本。我们把风险显露的损失差与风险减轻成本的比称为风险杠杆(risk leverage)。即

$$\text{风险杠杆} = \frac{\text{风险减轻前的风险显露} - \text{风险减轻后的风险显露}}{\text{风险减轻成本}}$$

风险监控

随时监控的必要性：由于风险是一些概率事件，它经常依赖于外部因素。在外部因素改变以后，风险构成的威胁可能和以前的评估有很大的差别。显然，对风险的理解也要随时间改变，进而所采取的风险化解措施可能影响着对风险的认识

跟踪监控：上述的风险动态特性表明，不应把项目的风险看成是静止不动的。必须定期对风险进行重新评估

进度管理

值得重视的现象

软件项目能否按计划的时间完成，及时提交产品是项目管理的一个重要课题。我们都希望按计划及时完成，但项目未能按预期的进度提交产品，延误工期的现象经常会出现。我们必须重视这一现象，分析其原因，并有针对性地采取措施

制订项目进度安排的条件

制订项目进度安排计划是为了实施，自然希望越准确，越符合实际越好，但是怎样才能做到这一点，需要在这以前做些工作，创造良好的条件，使得进度安排的确定是有根据的

进度控制问题

这些条件包括以下7条：

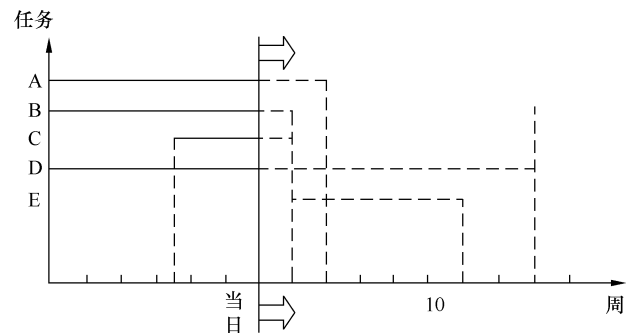
1. 项目分解
2. 确定各部分之间的相互关系
3. 时间分配
4. 确认投入的工作量
5. 确定人员的责任
6. 规定工作成果
7. 规定里程碑

甘特图 (Gantt chart)

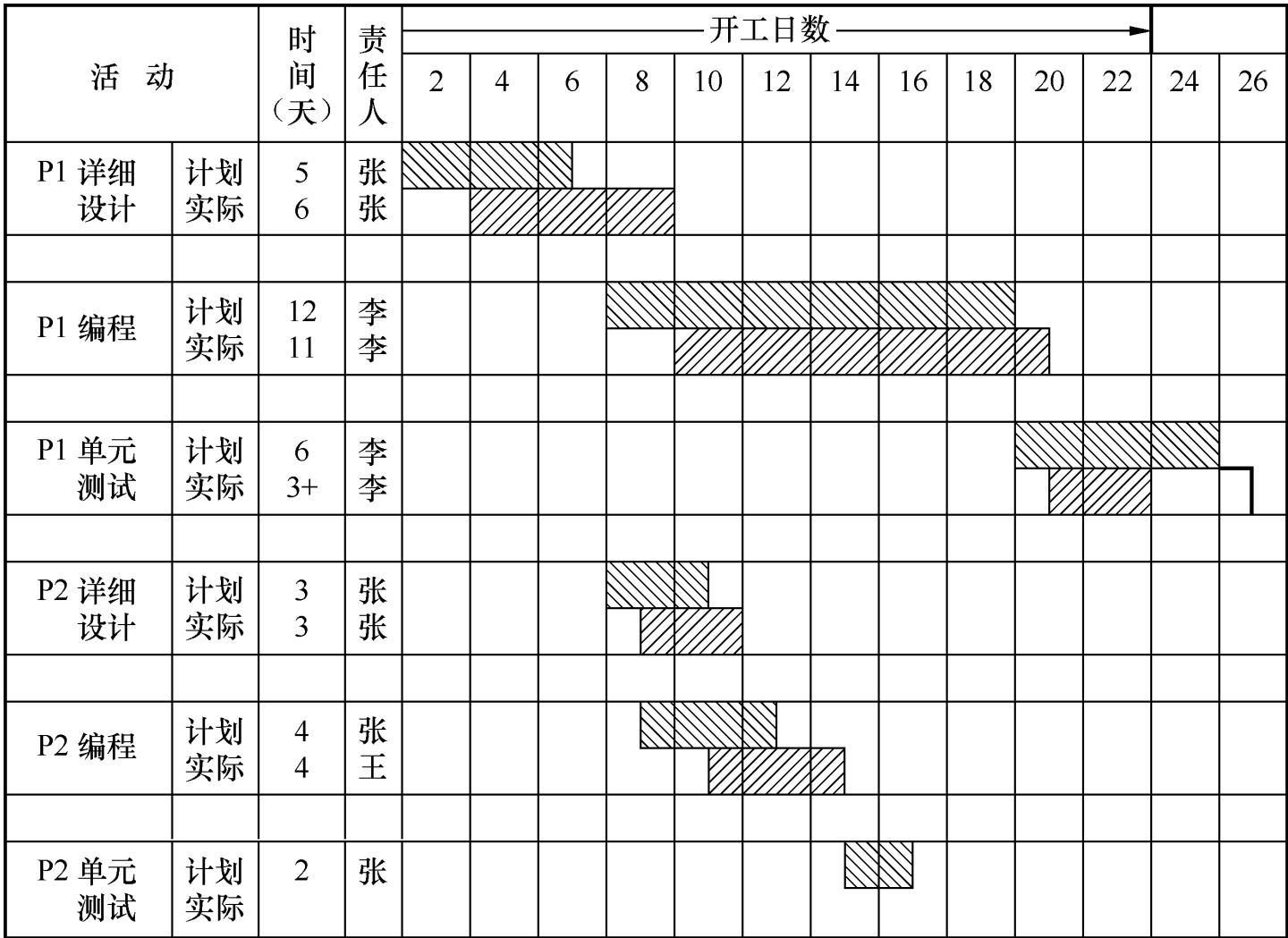
甘特图是表示工作进度计划以及工作实际进度情况最为简明的图示方法

甘特图中横坐标表示时间，以水平线段表示子任务的工作阶段，可以为其命名
线段的起点和终点分别对应着该项子任务的开工时间和完成时间，线段的长度表示完成它所需的时间，有实线和虚线之分，一开始做出各项子任务的计划时间，应该都以虚线表示

甘特图可以清楚地表示各项子任务在时间对比上的关系，但无法表达多个子任务之间更为复杂的衔接关系



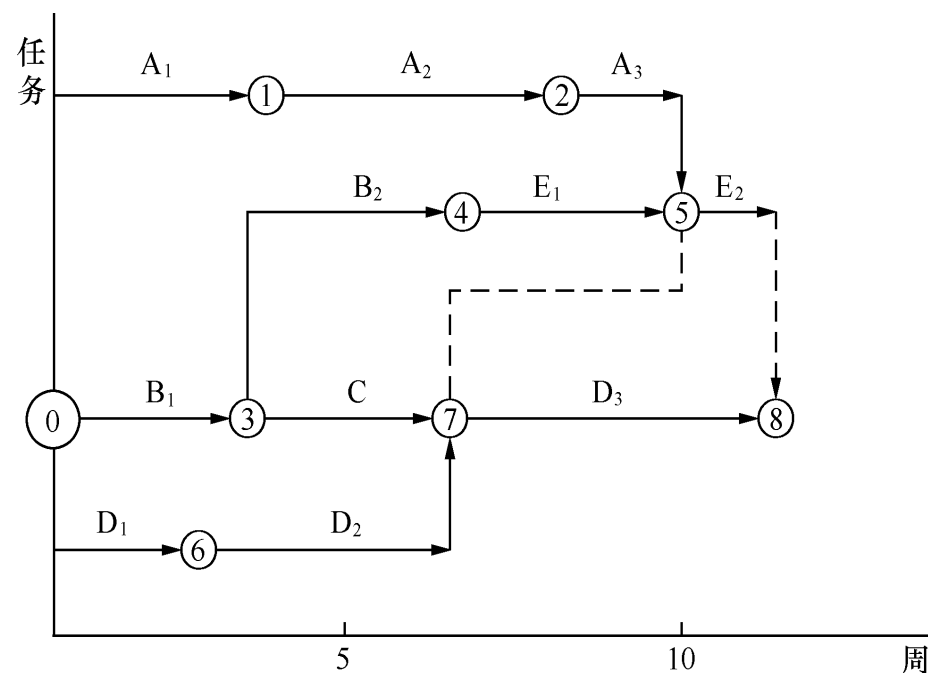
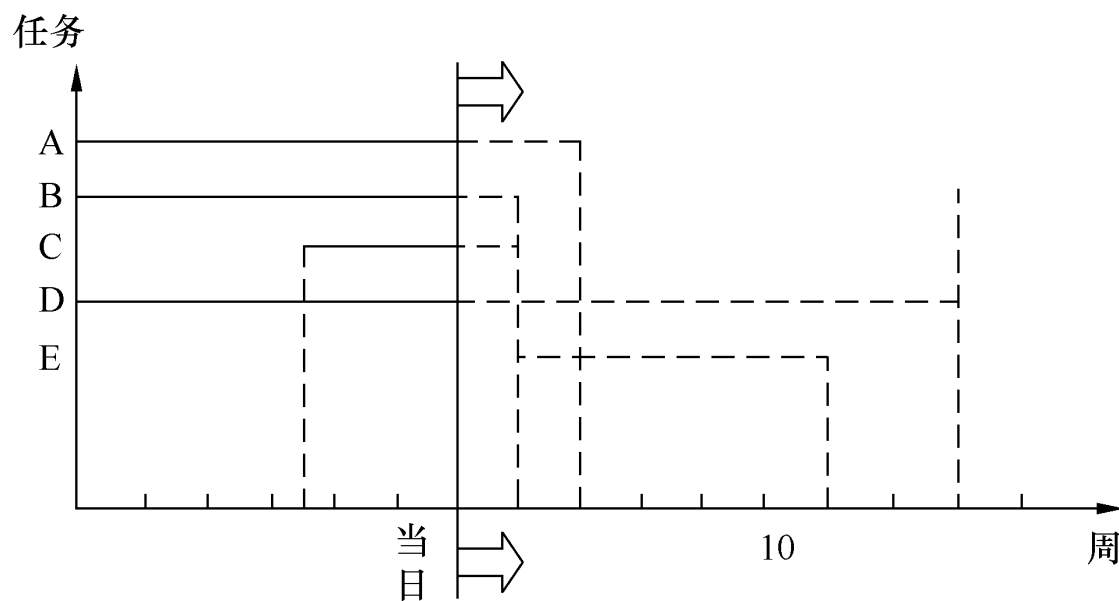
甘特图



时标网状图 (Time Scalar Network)

为克服甘特图的缺点，将甘特图做了一些修改，形成了时标网状图

图中的任务以有向线段表示，其指向点表示任务间的衔接点，并且都给予编号，可以显示出各子任务间的依赖关系。它显示出比甘特图具有优越性



PERT图

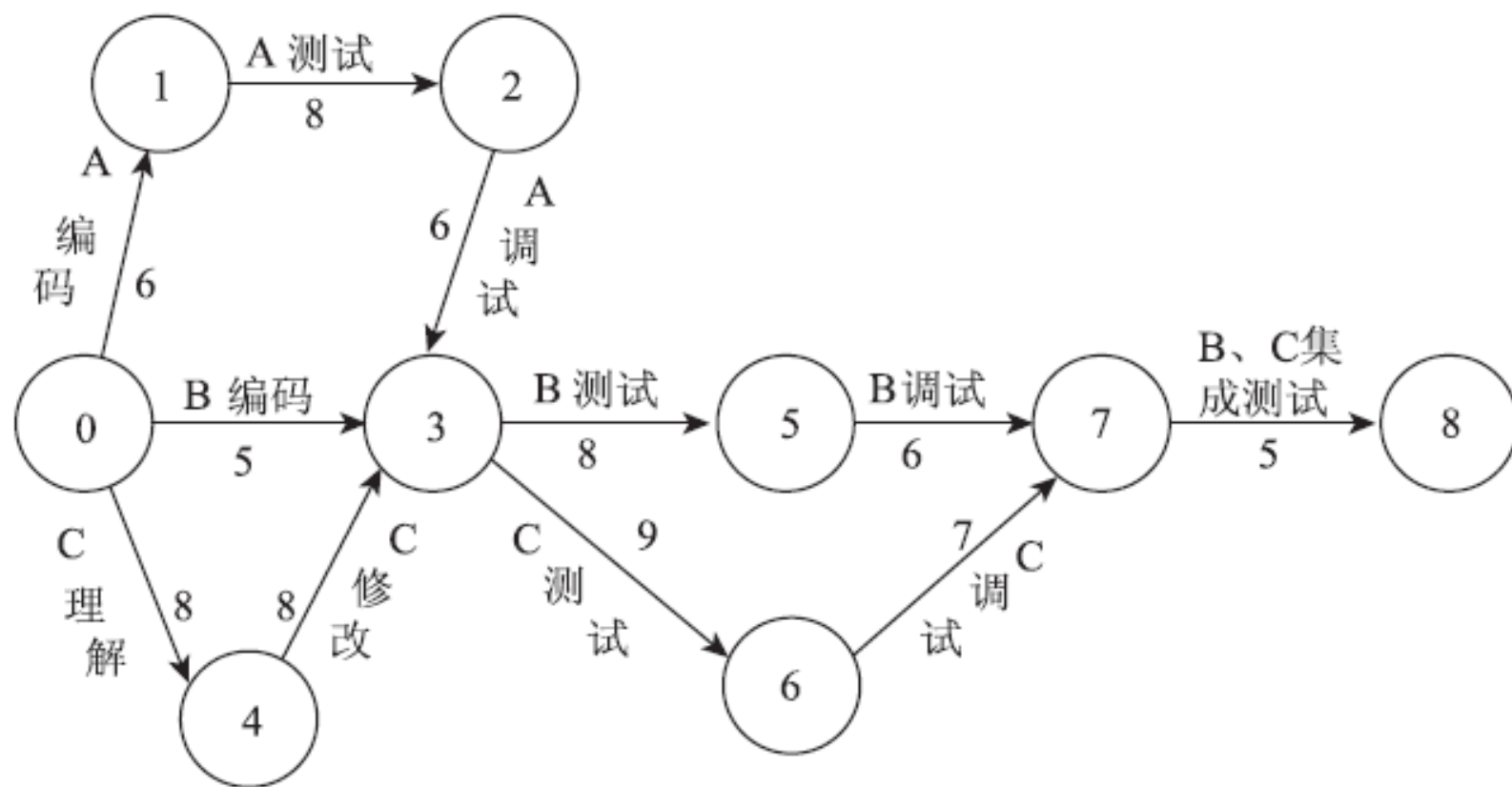
计划评审技术 (program evaluation and review technique, PERT) 也称网络图方法, 或简称PERT图方法, 它的另一名称是关键路径法 (critical path method, CPM)

PERT图中以有向的箭头作为边表示子任务, 它是有名称 (即子任务名)、有长度 (即完成此项子任务所需的时间) 的向量

以有编号的圆圈作为结点, 它应该是子任务向量的始发点或指向点

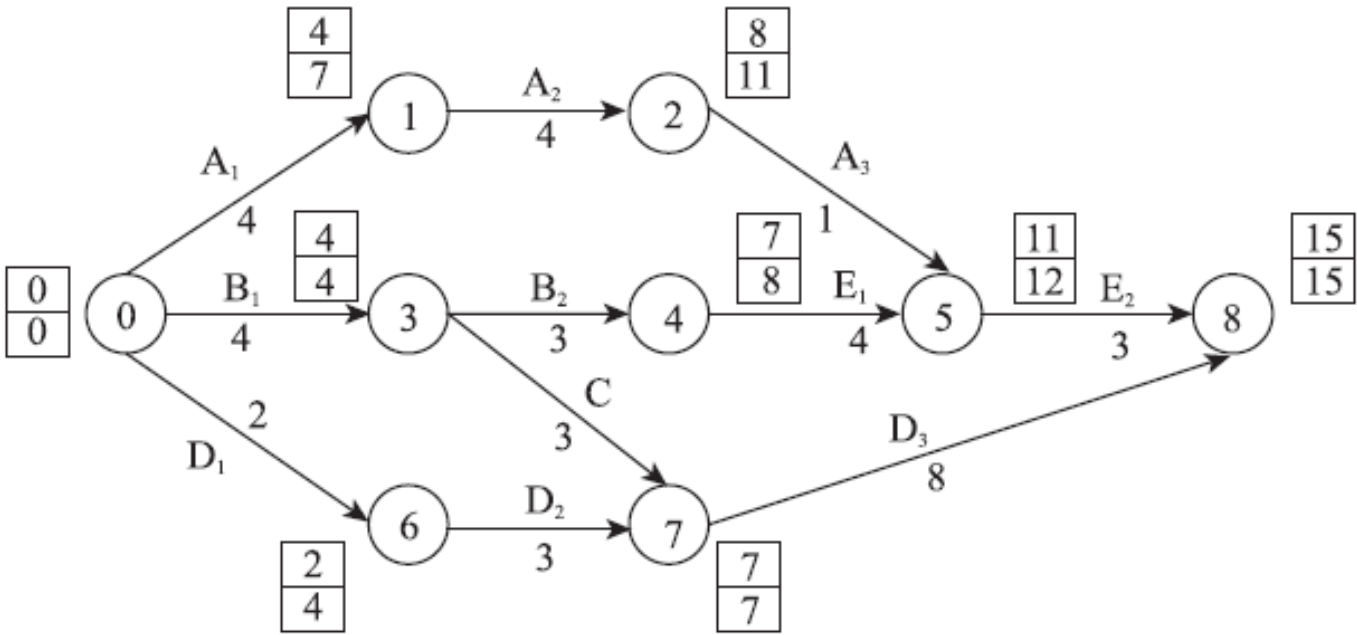
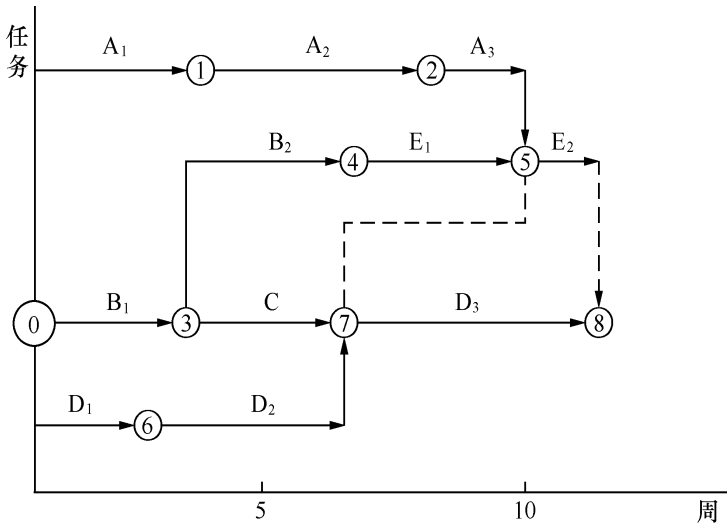
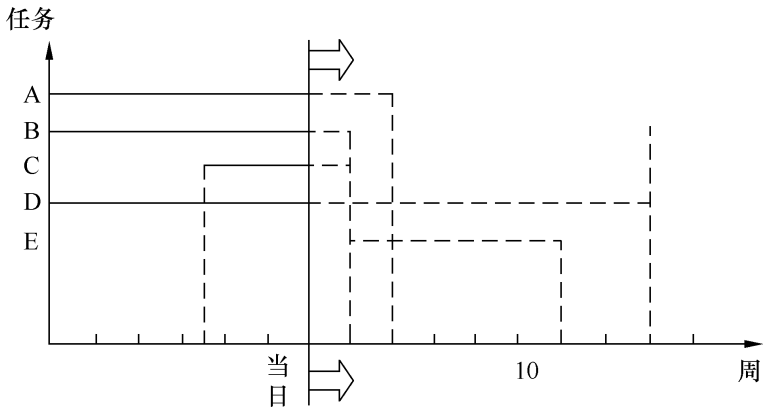
由若干条边和若干个结点构成了网状图, 于是我们可以沿相互衔接的子任务形成的路径, 进行路径长度的计算、比较和分析, 从而实现项目工期的控制

PERT图

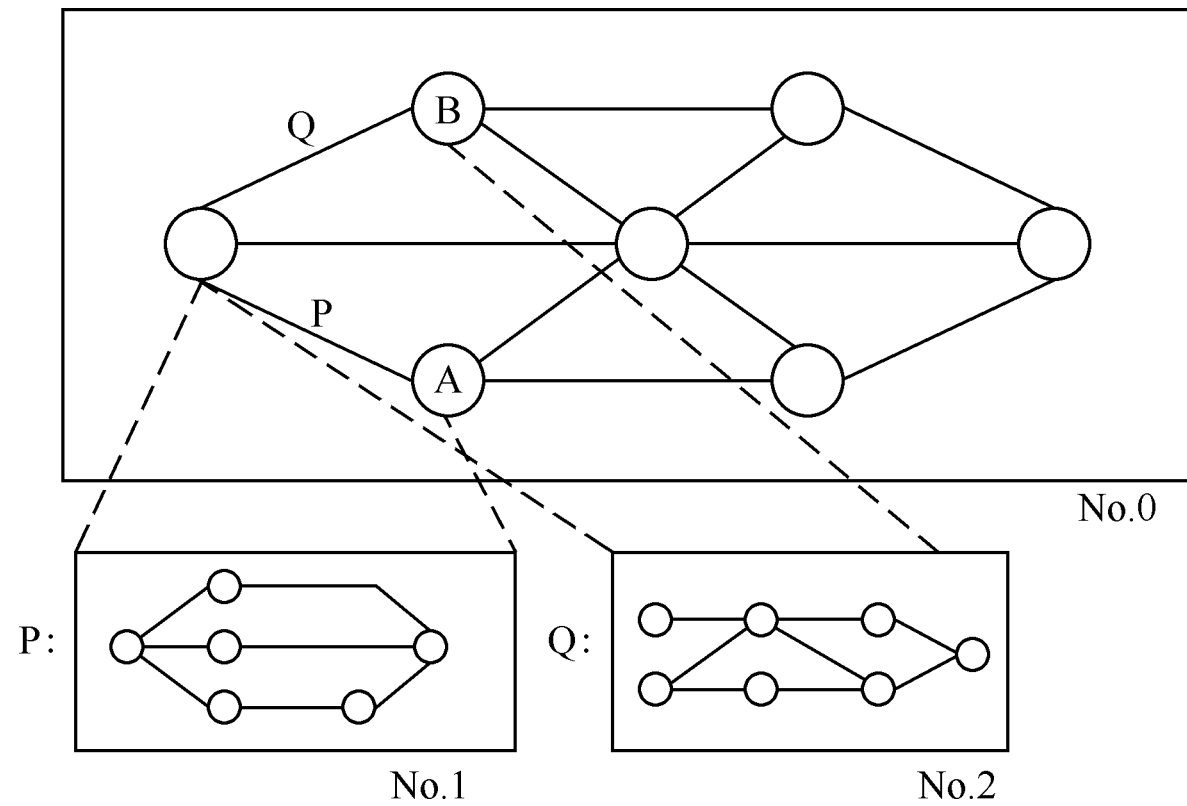


开发模块 A、B、C 的网状图

PERT图



分层PERT图



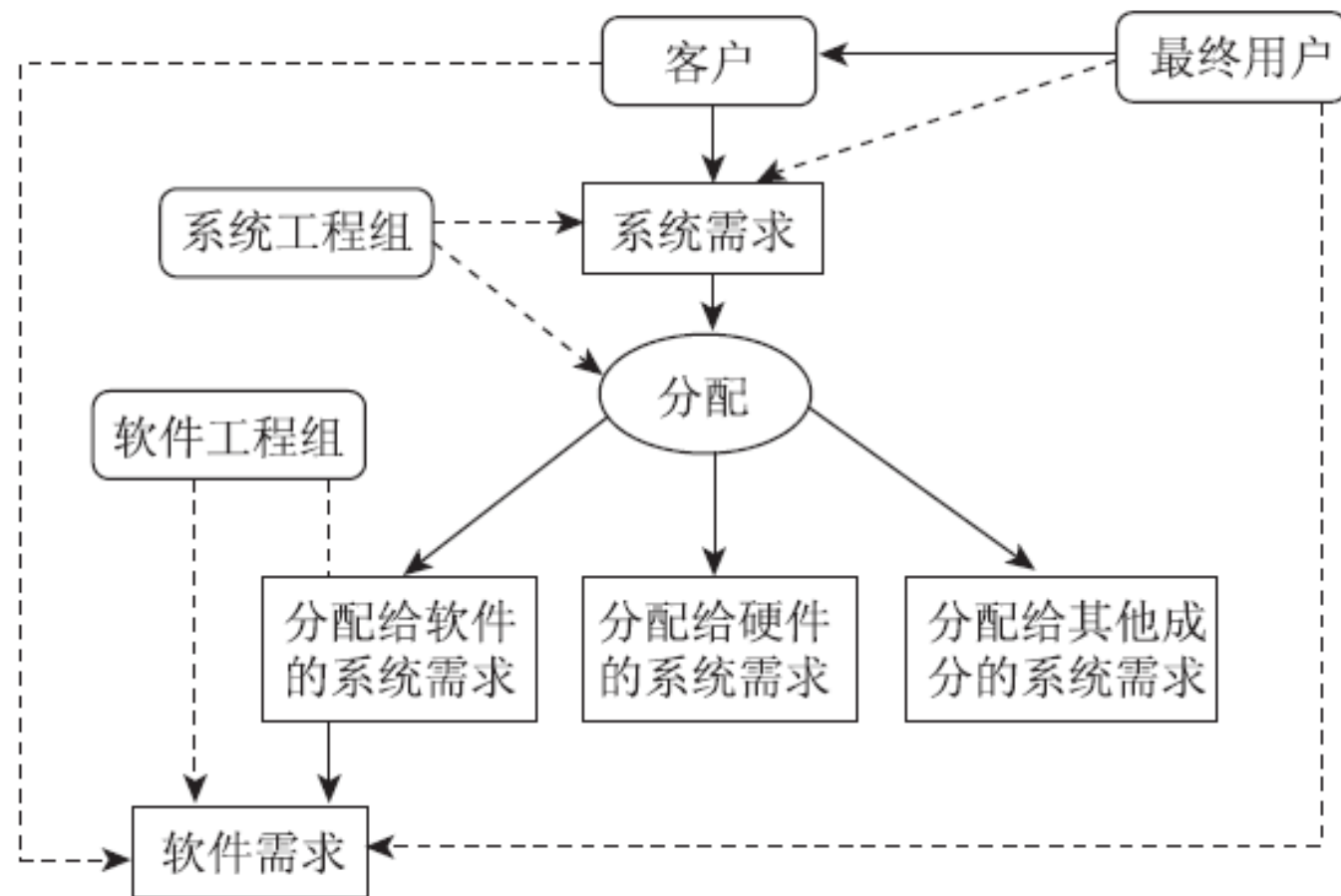
需求管理

系统和系统需求分配

系统：通常考虑的系统是指基于计算机的系统或计算机控制的系统，它是在计算机控制下完成特定功能的系统。例如，航空管制系统、飞行器惯性导航系统、生产控制系统等

系统需求分配：系统工程组面对用户，负有开发系统的责任。系统工程组在从用户那里取得系统需求以后，应将系统需求进行分解，也就是把已确定的系统需求分配给系统的各个组成部分

系统需求分配



系统需求分配

软件需求

按IEEE STD 610标准的定义，软件需求是用户为解决某个问题或为实现某个目标，要求软件必须满足的条件或能力

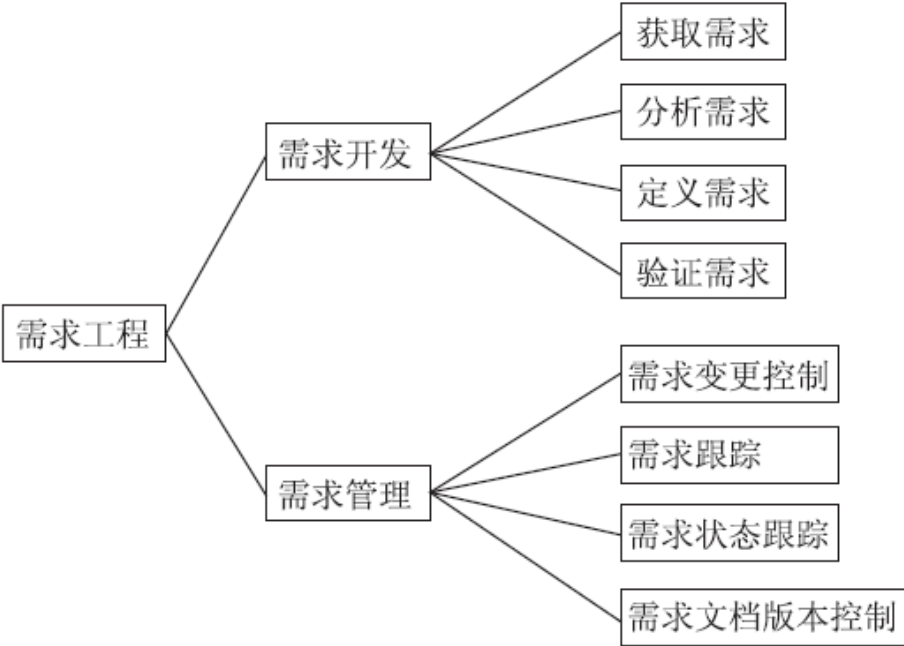
软件需求的3个层次如下：

- 1. 业务需求：客户对软件的高层目标要求**
- 2. 用户需求：用户使用软件必须达到的要求和完成的任务。通常在用例(use case)或方案脚本(scenario)中加以说明**
- 3. 功能和非功能需求：规定了开发人员必须实现的需求，它的实现将满足上述业务需求和用户需求。通常以需求规格说明(requirement specification)的形式给以详尽描述**

需求工程

需求工程是系统工程或软件工程中解决需求问题的一个崭新领域。其目标在于使得到的产品能够准确、真实地体现客户的需求，令客户满意

需求工程包括两个方面：需求开发与需求管理



需求工程的构成

需求变更

需求变更难于完全避免，系统需求或软件需求往往在开发工程中发生变更，提出变更有可能在开发的任何阶段

需求变更原因：

- 1. 单纯的用户因素**
- 2. 市场形势变化引发的需求变更**
- 3. 系统因素。在系统内部，如计算机硬件、系统软件或数据等的变更要求软件与其相适应**
- 4. 工作环境因素。与软件运行相关的工作制度或法规、政策的变更，或业务要求变更导致的需求变更**
- 5. 需求开发工作有缺陷，可能有两种情况：一是需求分析、定义和评审工作不够充分，致使需求规格说明中隐含着问题；二是需求开发中开发人员与用户沟通不够充分**

需求变更

需求变更对软件开发工作的影响：

- 1. 使得变更前的开发工作和成果失效**
- 2. 使得返工成为不得不采取的对策**
- 3. 势必带来软件开发计划的相应变更、开发成本的相应增加和开发工作量及资源投入的追加**

需求变更

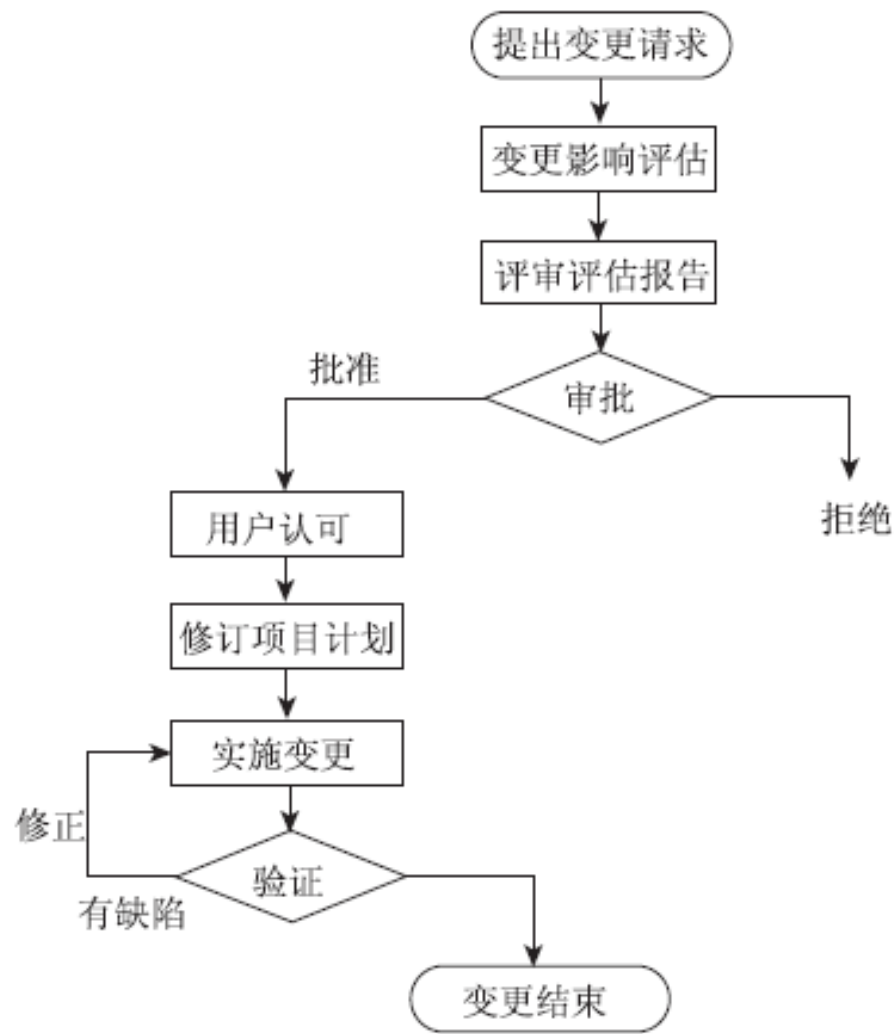
降低需求变更风险的策略：

1. 在需求开发工作中要与客户充分沟通
2. 与用户共同确定需求
3. 开发组织和用户双方签署的项目开发合同中应包括对出现需求变更的应对条款
4. 如果项目自身具有需求不易确定的特点，在项目启动时最好采用快速原型方法或螺旋模型，以便在确认需求的基础上开发产品
5. 在项目开始时，如估计到需求可能变更，则可在开发计划中适当留有余地，以防变更需求造成被动
6. 严格实施变更控制，使产品质量不致因需求变更受到影响

需求变更控制

变更控制的步骤

1. 提出变更请求
2. 审理变更请求，进行变更影响评估
3. 批准变更请求
4. 取得用户的认可
5. 修订项目计划
6. 实施变更
7. 验证变更



需求变更控制的流程

可追溯性管理

需求可追溯性与需求变更控制

- 1. 如果将笼统的需求状态演变概念加以具体化，考虑某一项特定的需求，它也必然随着开发工作的进展而逐步扩展和演化**
- 2. 如果以某种方式(例如以下给出的可追溯矩阵)对其做出确切的表达，那么需求变更无论出现在任何阶段，都能沿用着这一线索进行无遗漏的追踪，对相关部分实施修正和调整，最终做到变更控制**

可追溯性管理

可追溯性管理的目标

1. 实施需求可追溯性管理应使每一项需求均能追溯到，包括对应的设计、实现该项需求的代码以及测试此项实现的用例
2. 这样便可做到确保软件产品满足所有需求，并已测试了所有需求，从而使表现前后继承关系的脉络清晰可见

两类不同的追溯

1. 向前追溯：沿生存期从需求跟踪到设计、编码、测试等后继阶段所输出工作产品的相关元素
2. 向后追溯：从各阶段工作产品的元素反向追溯，直至追溯到初始需求

配置管理

软件工程项目随着工作的进展会产生多种信息，包括技术资料、管理资料等，如何管好这些资料是项目管理面临的重要问题

另一方面，还必须考虑到，这些资料和信息不仅不断地产生，而且还在不断地演化和变更

如何遵循一套严谨、科学的管理办法，使信息和资料的产生、存放、查找和使用既有序又高效，不致发生混乱和差错的现象，这正是配置管理所要解决的问题

配置管理

软件配置管理的目的是为某个过程或某个项目的软件项建立和保持完整性，以便相关方便于使用

软件配置管理要开展的活动包括：配置标识、配置控制、配置状态报告、配置评价以及发布管理、交付等

软件配置管理的对象称为软件配置项（software configuration item），包括：

- 1. 与合同、过程、计划和产品有关的文档及数据**
- 2. 源代码、目标代码和可执行代码**
- 3. 相关的产品，包括软件工具、库内的可复用软件、外购软件及顾客提供的软件**

软件配置管理的主要任务

1. 制订软件配置管理计划。包括：

- ① 配置标识规则
- ② 如何建立配置数据库，并将配置项置于配置管理之下
- ③ 配置管理人员的职责及配置管理活动
- ④ 所采用的配置管理工具、技术和方法

2. 实施变更管理，防止项目进行中因变更导致的混乱。

3. 实施版本管理和发布管理。

软件配置管理的工作是要解决下列问题

1. 采用什么方式去标识和管理数量众多的程序、文档等的各种版本？
2. 在软件产品交付用户之前和交付之后如何控制变更？实现有效的变更？
3. 谁有权批准变更以及安排变更的优先级？
4. 用什么方法估计变更可能引起的其他问题？

这些问题的解决正是软件配置管理应完成的任务：**配置标识、版本管理、变更管理、配置审核及配置报告**

本节内容

Readings

《软件工程概论》，郑人杰，马素霞等著，第十三章

Thanks