

软件工程

Software Engineering (18/11)



第一章 软件工程概述

穆海伦

计算机学院 智能与软件研究所

E-mail: helen_se@163.com QQ: 1055874556 PH: 13750802617(612617)



软件工程

1

什么是软件

2

软件危机

3

软件工程

4

软件过程

5

软件工程知识体系SWEBOK

6

项目管理概述



1. 什么是软件

软件是计算机系统中与硬件相互依存的另一部分，它包括程序、数据及相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序能正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文材料。

什么是软件？（站在软件工程的角度看）

软件就是：

- ❑ 一个或多个计算机程序，其执行时能提供所期望的功能和性能
- ❑ 一个或多个数据结构，这些结构使得程序能够完全操纵信息
- ❑ 一个或多个文档，这些文档描述了程序分析、设计、实现和维护的细节





- ❑ 50年代：软件=程序
- ❑ 60年代：软件=程序+文档（分析、设计、测试、维护，但不包括管理文档）
- ❑ 70年代：软件=程序+文档+数据（初始化数据、测试数据、研发数据、运行数据、维护数据、工程数据、项目管理数据等）
- ❑ 1984年美国开始认识到软件管理是一个过程管理，1991年出现CMM1.0，1996年出现了UML。
- ❑ “软件工作产品”——开发过程中产生的各种软件
- ❑ “软件产品”——最后交付的软件

从这里，我们开始感到，软件——这个我们追逐的对象，已经逐渐地不再成为软件经理的唯一目标，我们开始更多地关注产生这个目标的过程。走完这个思路过程，花费了软件人30年以上的时间。





- ❑ 软件是一种逻辑实体，而不是具体的物理实体。
抽象性：可记录，但看不到
- ❑ 软件的生产与硬件不同。
 - 无明显的制造过程
 - 软件是开发出来的，不是制造出来的
- ❑ 在软件的运行和使用期间，没有硬件那样的机械磨损、老化、折旧等问题。
软件可能被“废弃”，但不会“用坏”
- ❑ 软件的开发和运行常常受到计算机系统的限制，对计算机系统有着不同程度的依赖性。
- ❑ 软件的开发至今尚未完全摆脱手工艺的开发方式。
- ❑ 软件是复杂的。（软件复杂性来源于它所反映的实际问题的复杂性。）
- ❑ 软件成本相当昂贵。
 - 软件大部分是定制的，而不是装配的
 - 可复制性：与开发成本相比，复制成本很低，存在软件产品的保护问题
- ❑ 相当多的软件工作涉及到社会问题。



- ❑ 按功能分：系统软件、支撑软件、应用软件
- ❑ 按规模分：大型、中型、小型
- ❑ 按工作方式分：实时/分时、交互/批处理
- ❑ 按服务对象分：定制软件、产品软件
- ❑ 按销售方式分：定单软件、非定单软件





2. 软件危机

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。这些问题绝不仅仅是不能正常运行的软件才具有的，实际上，几乎所有软件都不同程度地存在这些问题。

概括地说，软件危机包含下述两方面的问题：

- ❑ 如何开发软件，以满足对软件日益增长的需求；
- ❑ 如何维护数量不断膨胀的已有软件。





软件危机表现在软件的可靠性没有保障、维护费用不断上升、进度无法预测、成本增长无法控制、程序员无限度增加等，形成软件开发局面失控的状态。

具体地说，软件危机主要有以下一些典型表现：

- ❑ 对软件开发的进度和成本无法估计
- ❑ 用户对已经开发完成的软件的满意度非常低
- ❑ 软件质量无法保证
- ❑ 软件开发后的维护工作很难进行
- ❑ 软件通常没有合适的文档资料
- ❑ 软件成本在系统总成本中所占的比例越来越高
- ❑ 软件开发生产率提高的速度，远远跟不上计算机应用迅速普及深入的趋势。





软件危机的例子很多，经常被拿出来作为典型的有：

- ❑ 1962年美国水手 I 号因导航软件一个语句的语义错误，导致偏离航线，任务失败。
- ❑ 阿波罗8号因计算机软件错误，造成存储器信息丢失。
- ❑ 阿波罗14号在飞行的10天中，出现了18个软件错误。
- ❑ 美国IBM公司的OS/360系统，花了几千人很多年的努力而失败。





与人类社会其他产品发展的历史比较，可以看出，软件开发的历史具有一些独特的发展特性：

- ❑ 与建筑技术、制造技术、计算机硬件技术不同，软件生产具有非常特殊的个体化、个性化和生产过程的不可复制性。
- ❑ 软件开发的工具、技术手段获得了很大的进步，但开发方法、管理方法和水平并没有获得自动和同步的进步。

在那个年代，软件开发团队通常被视为“软件作坊”，主要表现在：

- ❑ 个人对所负责的“局部”负责、在这个局部是完全个性化和自由的，系统就是由几个这样的“局部”构成的叠加（例如：通过菜单的形式）构成的。
- ❑ 没有任何设计文档和可用于维护的资料，即生产过程是没有记录的。
- ❑ 没有评审和独立的系统测试，即结果不依靠第三方确认和验证。
- ❑ 进度、成本、质量是不可预测的。

而更要命的是，同处于计算机“生物圈”的计算机硬件，却有一个可怕的摩尔定律：硬件成本每隔18个月就降低一半，例如：存储器每年降低40%、主机硬件的性价比每十年提高一个数量级。而这个定律直到现在也没有被打破。可想而知，软件人应该是怎样的郁闷和压抑



软件危机产生的原因

在软件开发和维护的过程中存在这么多严重问题，

- ❑ 一方面与软件本身的特点有关。
- ❑ 另一方面也和软件开发与维护的方法不正确有关。





为了消除软件危机：

- ❑ 首先应该对软件有一个正确的认识。应该彻底消除在计算机系统早期发展阶段形成的“软件就是程序”的错误观念。一个软件必须由一个完整的配置组成。
- ❑ 更重要的是，必须充分认识到软件开发不是某种个体劳动的神秘技巧，而应该是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目。
- ❑ 必须充分吸取和借鉴人类长期以来从事各种工程项目所积累的有效的原理、概念、技术和方法，特别要吸取几十年来人类从事计算机硬件研究和开发的经验教训。
- ❑ 应该推广使用在实践中总结出来的开发软件的成功的技术和方法，并且研究探索更好更有效的技术和方法，尽快消除在计算机系统早期发展阶段形成的一些错误概念和做法。
- ❑ 应该开发和使用更好的软件工具。

总之，为了解决软件危机，既要有技术措施(方法和工具)，又要有必要的组织管理措施。软件工程正是从管理和技术两方面研究如何更好地开发和维护计算机软件的一门新兴学科。





60年代软件生产的特征是：软件作坊——个体化生产。

为了跟上硬件的发展速度、解决进度问题和成本可控制、质量可保证、系统可维护、过程可管理等问题，产生了把软件开发，作为一个工程来管理的思想，即软件工程的

软件工程的提出：

- ❑ 为摆脱软件危机，北约（NATO）的科学委员会于1968年在联邦德国召开德有关研讨会上，首次提出了软件工程（Software Engineering）的概念，其主要想法，是把人类长期从事的其他领域的各种工程所积累的经验行之有效的原理、概念、技术和方法，包括对硬件研发积累的经验，导入到软件开发中。
- ❑ 到70年代末，已经取得了大量的研究成果，形成了基本的方法，这一代的软件工程，称为“传统软件工程”。



概括地说，软件工程是指导计算机软件开发和维护的一门工程学科。采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，以经济地开发出高质量的软件并有效地维护它，这就是软件工程。





IEEE计算机学会将“软件工程”定义为：

- ① 应用系统化的、学科化的、定量的方法，来开发、运行和维护软件，即，将工程应用到软件。
- ② 对①中各种方法的研究。

(参 见 ： IEEE Standard Glossary of Software Engineering Terminology.IEEE, Piscataway,NJ std 610.12-1990, 1990)

把这个定义分解开来理解，就是：

- ❑ 应用计算机科学、数学及管理科学等原理，借鉴传统工程的原则和方法，来创建软件，从而达到提高质量、降低成本的目的。
- ❑ 其中，采用的方法包括：
 - 计算机科学和数学用于构造模型、分析算法；
 - 工程科学用于制定规范、明确风险、评估成本和确定权衡；
 - 管理科学用于进度、资源、质量、成本的管理。

因此，软件工程是计算机科学、工程和管理三个学科的综合。





1968 年正式提出“软件工程”这一术语之后，软件工程的专家和学者围绕计算机科学、工程和管理三个方面，做了很多研究，建立了早期关于软件工程管理的一些基本准则，从中可以看出早期软件工程的一些思路与出发点。

其中最著名的是著名软件工程专家B.W.Boehm在1983 年的一篇文章中，提出的软件工程7条基本原理，反映了作为软件工程应该关注和考虑的若干本质问题：

- ☐ 用分阶段的生命周期计划严格管理。
- ☐ 坚持进行阶段评审。
- ☐ 实行严格的产品控制。
- ☐ 采用现代的程序设计技术。
- ☐ 结果应能清楚地审查。
- ☐ 开发小组的人员应该少而精。
- ☐ 承认不断改进软件工程实践的必要性。





1968年正式提出“软件工程”这一术语之后，软件工程经历了以下几个发展阶段：

- ❑ 到70年代末，按照软件工程的思路，已经取得了大量的研究成果，形成了基本的方法，这一代的软件工程，称为“**传统软件工程**”。传统软件工程最主要的成果，就是所谓的软件生命周期的“瀑布模型”和相应生命周期的工作规范。从软件工程的生命周期划分及任务定义中，我们可以明显地看到上述软件工程基本原则的实际应用。
- ❑ 80年代以后，面向对象的方法和技术受到广泛的重视，软件工程的重点，转向OO。以Smalltalk-80语言的出现，标志着面向对象的程序设计与分析，进入实用化阶段，演化成一套完整的软件开发方法和系统设计体系。一般把面向对象的设计，称为软件工程的第二代，也称为**面向对象的软件工程**。
- ❑ 80年代中期，人们在研究和实践中发现，为了提高软件生产率，并使软件质量得到保证，其关键在于软件开发和维护中的管理和支持能力，这其中的关键，是所谓“软件过程”。从1984年开始，掀起了“软件过程运动”，91年出现CMM，是软件过程的典型代表。以后，逐渐形成了“**软件过程工程**”，这可以称为软件工程的第三代。
- ❑ 进入90年代以后，软件工程的一个重要进展，就是基于构件的开发。尽可能地利用可复用的构件，组装成新的系统，提高了软件的生产率、减少了故障和成本。基于构件的系统，更适应了Internet技术和分布式系统开发的需要。因此，有人把基于构件的软件开发，称为**第四代软件工程**。





传统软件工程把软件的生命周期定义为6个阶段：

❑ 问题定义与可行性研究：

指系统分析员通过对系统实际用户、使用管理部门、相关部门及人员进行的实际调查，搞清楚“问题”的背景、目的是什么？然后，据此提出关于“问题”的性质、工程目标、规模、相关联系等项目的基本情况，进行可行性分析，编制开发计划。

❑ 需求分析：

是通过问题识别、分析与综合、制订规格说明和评审等阶段，达到以下一些需求分析阶段的目标，它们都是对“用户需求”进行更专业化的“描述”与“转换”。需求分析的任务和目标包括：

- （1）确定对系统的综合要求；（2）分析系统的数据要求；
- （3）抽象出并确立目标系统的逻辑模型；（4）编写需求规格说明书。

❑ 软件设计：

软件设计包括概要设计和详细设计二个阶段：

➤ 在概要设计（总体设计）阶段，开发人员要回答需求分析中获得的系统目标如何去实现的问题，其目标是：

- （1）概要设计要体现对需求的完整实现；（2）概要设计要保证与需求的一致性；
- （3）概要设计能够达到向需求的反向可追踪；（4）概要设计关注系统结构设计的逻辑性、合理性和可扩展性；

传统软件工程提出了很多设计方法，最主要的面向结构的概要设计方法，包括：结构分析（SA）和结构设计（SD）等。

➤ 在详细设计阶段，是对概要设计进行细化，回答如何具体实现系统目标的问题。详细设计是面向具体程序编码，重点是编码规范。传统软件工程开发了HIPO（层次图加输入/输出处理）、PDL（过程设计语言）等详细设计工具。

❑ 编码：这个阶段的主要任务，是写出符合规范的代码，并完成相应的调试。

❑ 测试：测试包括：单元测试、集成测试、系统测试和验收测试等。

❑ 运行与维护：运行是把开发完成的系统，交付并达到正式使用的过程，维护包括：改正性维护、适应性维护、完善性维护和预防性维护等。



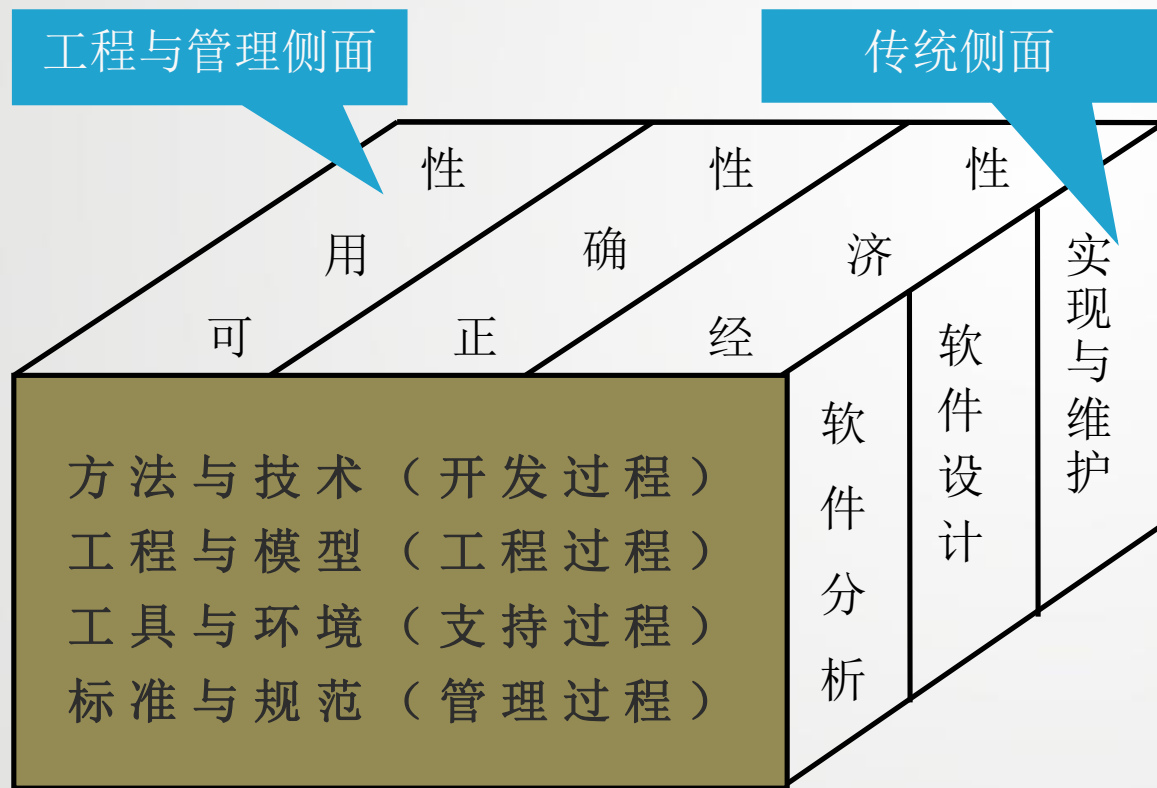


软件在发展，软件工程也在发展中，现代软件工程是在传统软件工程基础上发展起来的，既是传统软件工程概念和技术的自然演化，又具有区别于传统软件工程的独有的基本特征。现代软件工程已经比较明确自己的学习对象和目标，这就是：软件的工厂化生产。



现代软件工程更好地体现了“软件工程是计算机科学、工程与管理学科的结合”这一软件工程的定义和根本宗旨，因此，计算机科学、工程学和管理科学成为现代软件工程的主要知识来源和应用领域。

为了说明这三在者的关系，我们把软件工程看成是如下的一个“魔方”：



- ❑ 传统的侧面：包括需求分析、系统设计、实现与维护等软件开发的基本过程，我们称为“传统软件工程的基本过程”。
- ❑ 工程 and 管理的侧面：软件开发作为一个“工程”，必须满足可用性、正确性和经济性等基本工程要求，这是软件工程作为“工程”的方面。
- ❑ 在这样的组合下，软件工程可以被概括成4个过程，即：
 - 过程与模型
 - 方法与技术
 - 工具和环境
 - 标准和规范

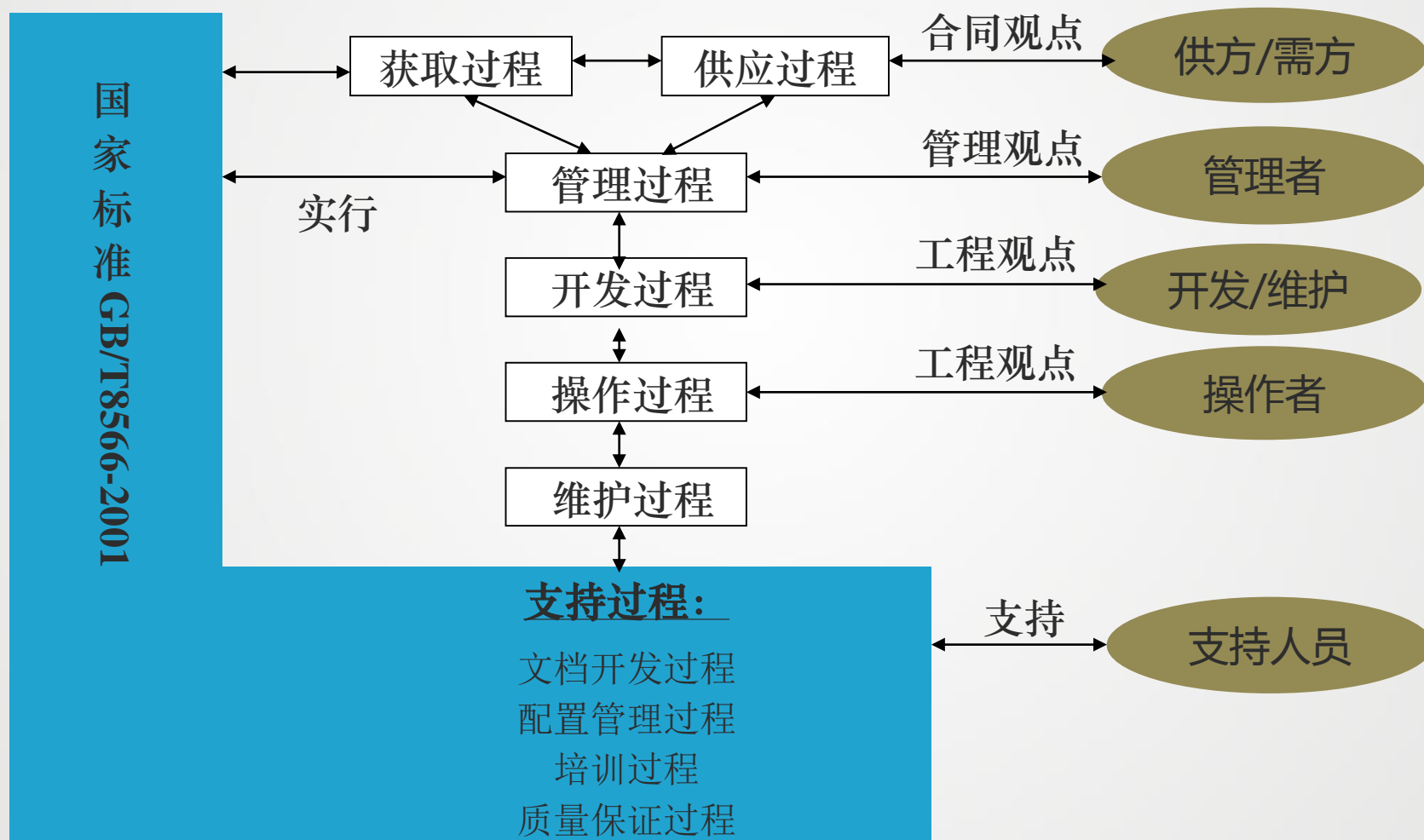




现代软件工程的软件生存周期

21

中国国家标准（GB/T8566-2001）将软件工程的生命周期划分成7个过程，基本上覆盖了开发、支持、工程及管理4个过程。





国标（GB/T8566-2001）软件工程生命周期7个过程的含义如下：

□ 获取过程：

需方按合同获取一个系统、软件产品和服务的活动，活动从定义软件产品或服务的获取需求开始，然后是准备并公布标书、选择供方和管理获取过程，直到系统的验收。

□ 供应过程：

供方向需方提供合同中的系统、软件产品和服务的活动，该过程的开始方法有二种：

- 一是准备一份建议书以应答需方的标书（定制系统）；
- 二是展示一个含有需方要求功能的软件系统（产品或服务）；

与需方签订合同或协议，供应过程规定了为管理和保证项目质量所需的步骤和资源，其中包括：制订项目计划和实施计划，直到向需方交付系统、产品或服务。

□ 管理过程：

按照管理的观点，一个机构（供方、需方、开发者、操作者和维护者）管理着各自的过程。管理过程定义了生存周期过程中的各项管理活动，包括：

- 项目的开始和范围定义
- 项目管理计划以及实施和控制
- 产品的评审和评价以及项目的完成。

在工程的观点下，开发者、操作者、维护者分别通过开发、操作、维护过程生产软件产品或提供服务

□ 开发过程：

开发过程是开发者为了定义和开发软件产品或服务所需要的活动，包括：需求分析、设计、编码、集成、测试、软件安装和验收等活动。

□ 操作过程：此过程定义操作者为了在规定的运行环境中为其用户运行一个计算机系统所需要的活动。

□ 维护过程：此过程定义维护者为了管理软件的更新、使其保持良好运行所需要的活动，包括系统的移植和退役。

□ 支持过程：支持过程对项目生存周期过程给予支持，有助于项目的成功并提高项目的质量。

- 各过程可以根据实际需要，进行裁剪或增加。
- 以上的几个过程，并不是就只在某一环节起作用，过了这个环节，该过程就结束了，它们是贯穿始终、协同工作的。





软件过程是一个为建造高质量软件所需完成的任务的框架，即形成软件产品的一系列步骤，包括中间产品、资源、角色及过程中采取的方法、工具等范畴。

软件过程(Software Procedure)是指软件生存周期所涉及的一系列相关过程。过程是活动的集合；活动是任务的集合；任务要起着把输入进行加工然后输出的作用。活动的执行可以是顺序的、重复的、并行的、嵌套的或者是有条件地引发的。

- ❑ 过程定义了运用方法的顺序、应该交付的文档资料、为保证软件质量和协调变化所需要采取的管理措施，以及标志软件开发各个阶段任务完成的里程碑。为获得高质量的软件产品，软件过程必须科学、有效。
- ❑ 没有一个适用于所有软件项目的任务集合。因此，科学、有效的软件过程应该定义一组适合于所承担的项目特点的任务集合。通常，一个任务集合包括一组软件工程任务、里程碑和应该交付的产品。
- ❑ 通常使用生命周期模型简洁地描述软件过程。生命周期模型规定了把生命周期划分成哪些阶段及各个阶段的执行顺序，因此，也称为过程模型。





所谓软件过程模型就是一种开发策略，这种策略针对软件工程的各个阶段提供了一套范形，使工程的进展达到预期的目的。对一个软件的开发无论其大小，我们都需要选择一个合适的软件过程模型，这种选择基于项目和应用的性质、采用的方法、需要的控制，以及要交付的产品特点。

一个软件过程模型是软件过程的一个抽象表示法。每个过程模型从一个特定的角度表现一个过程，只提供过程的某一侧面的信息。

典型的软件过程模型有：

- ☐ 瀑布模型
- ☐ 原型（迭代）模型
- ☐ 喷泉模型
- ☐ 增量模型
- ☐ 螺旋模型
- ☐ 构件组装模型





瀑布模型 (waterfall Model)

25

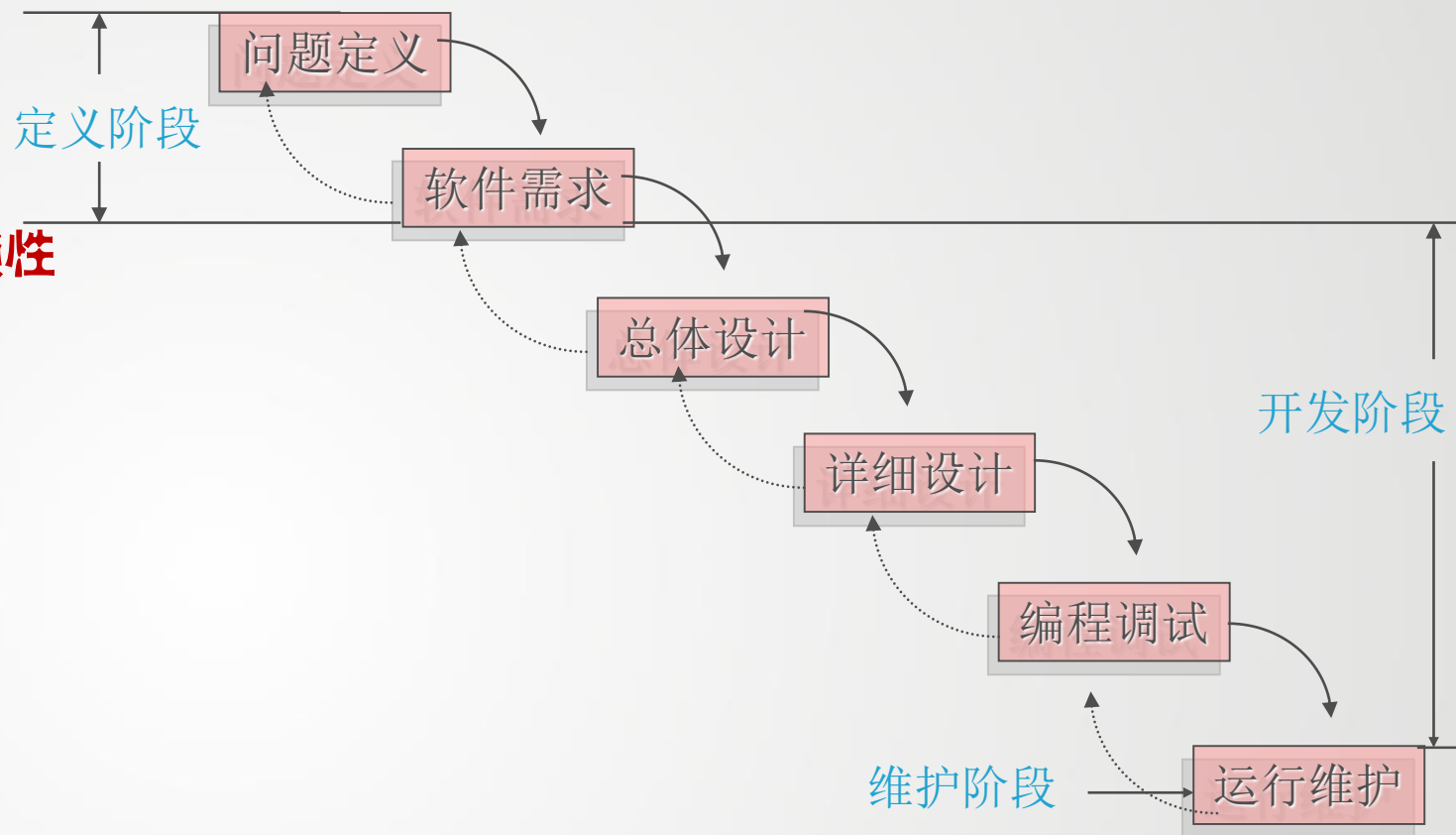
传统的生命周期模型

70年由Royce提出

典型瀑布模型具有顺序性和依赖性

瀑布模型的特征:

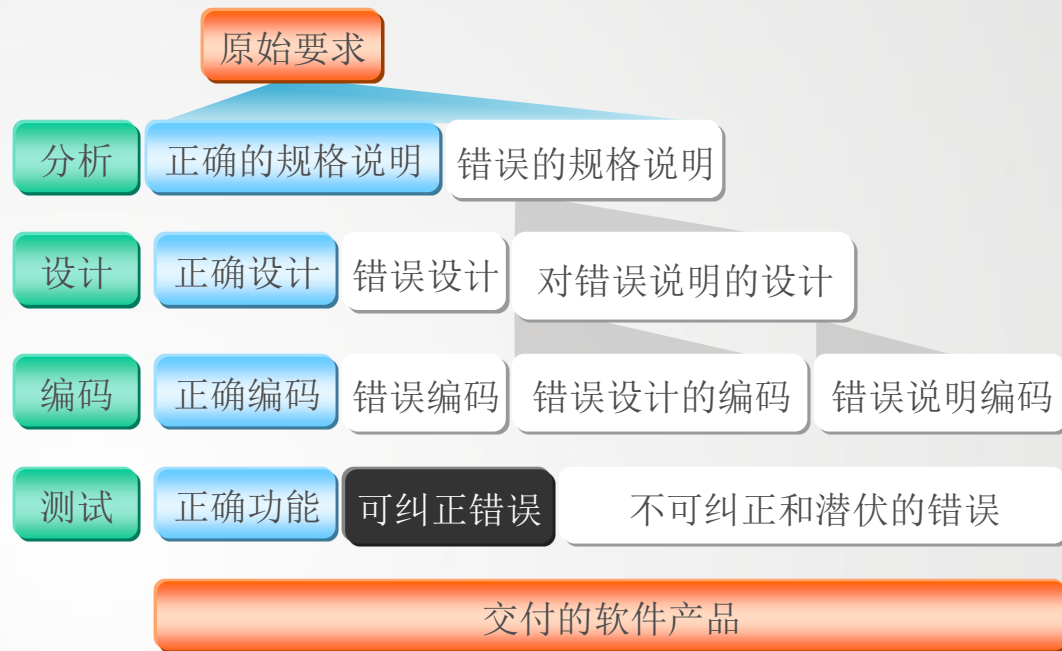
- ❑ 从上一项活动中接受该项活动的工作成果（工作产品），作为输入。
- ❑ 利用这一输入实施该项活动应完成的内容
- ❑ 给出该项活动的工作成果，作为输出传给下一项活动
- ❑ 对该项活动实施的工作进行评审。若其工作得到确认，则继续下一项活动。





瀑布模型的优点：

- ❑ 从上一项活动中接受该项活动的工作成果（工作产品），作为输入。
- ❑ 强调开发的阶段性；
- ❑ 强调早期计划及需求调查；
- ❑ 强调产品测试。



瀑布模型的缺点：

- ❑ 从认识论角度看，人的认识是一个多次反复循环的过程，不可能一次完成。但瀑布模型中划分的几个阶段，没有反映出这种认识过程的反复性。特别是瀑布模型过于依赖早期进行的唯一一次需求调查，不能适应需求的变化；
- ❑ 软件开发是一个知识密集型的开发活动，需要相互合作完成，但瀑布模型没有体现这一点。特别是由于瀑布模型是单一流程，开发中的经验教训不能反馈应用于本产品的过程。





原型（迭代）模型

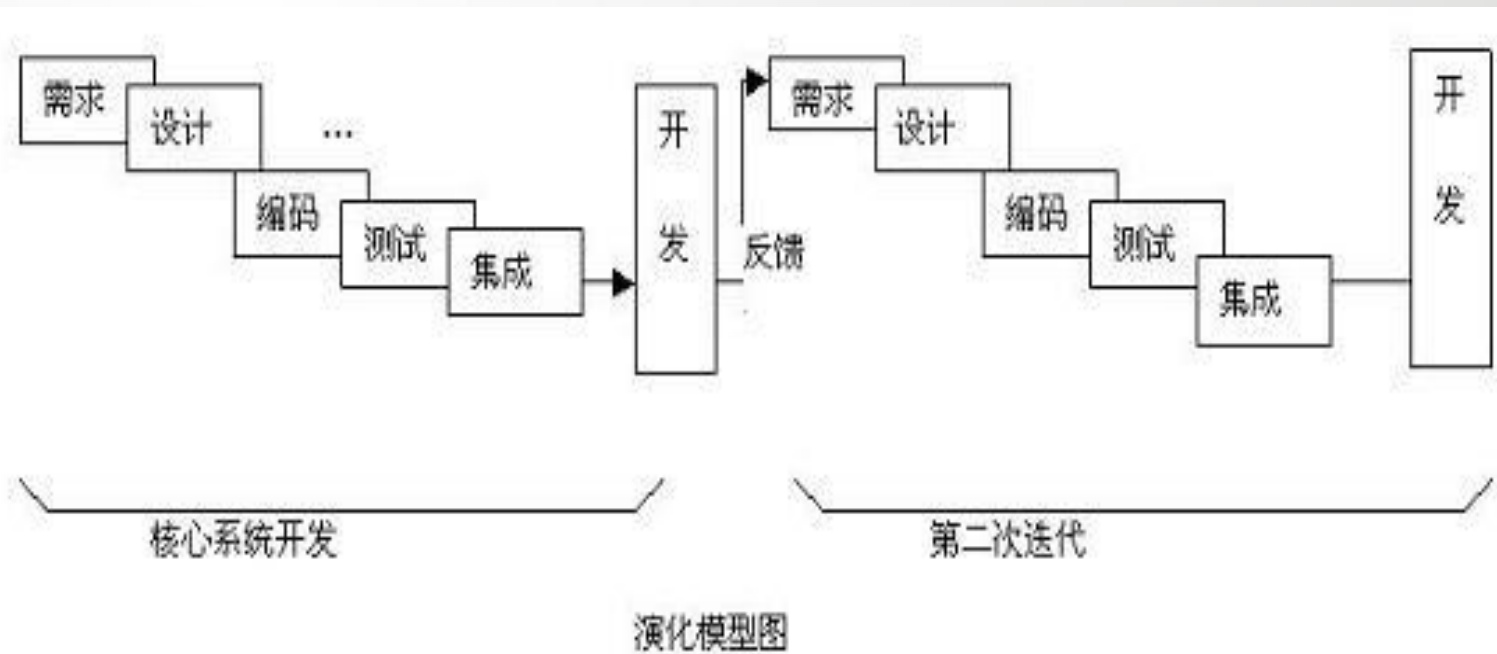
27

基本思想：

- ❑ 在获取一组基本的需求定义后，利用高级软件工具的可开发环境，快速地建立一个目标系统的最初版本-原型，并把它交给用户试用、补充和修改，再进行新的版本开发。反复进行这个过程，直到得出系统的“精确解”，即用户满意为止。
- ❑ 经过这样一个反复补充和修改的过程，应用系统的“最初版本”就逐步演变为系统的“最终版本”。
- ❑ 原型：一个真实的可执行模型，它实现了系统的若干基本功能。
- ❑ 原型法：不断地运行系统“原型”来进行启发、揭示和判断的系统开发方法。

来源：

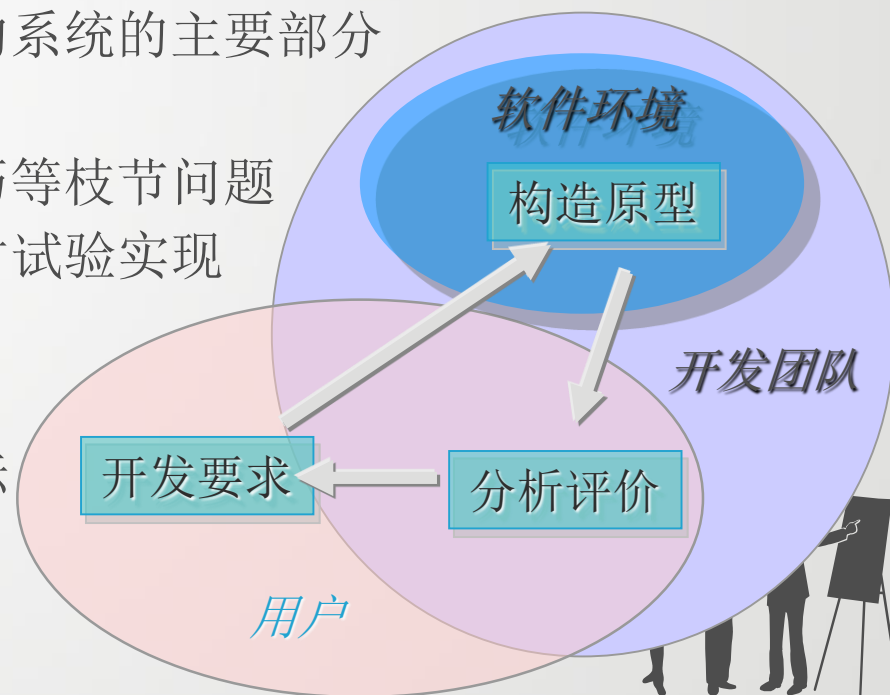
- ❑ 利用相应原型开发工具快速开发。
- ❑ 已经完成核心产品开发和试点，进行本地化、客户化的过程，称为二次开发（迭代和演化）。





快速建立原型的目的是，是获取需求：

- ❑ 在“需求分析”阶段，开发团队和用户一起为想象中的系统的某些主要部分，定义需求和规格说明，并由开发团队在规格说明级用原型描述语言构造一个系统原型，它代表了部分系统，包括哪些为满足用户需求的必要属性。
- ❑ 该原型可用来帮助分析和设计工作，而不是一个软件产品。
- ❑ 通过演示原型，用户可以根据他所期望的系统行为来评价原型的实际行为。如果原型不能满意地运行，用户能立刻找出问题和不可接受的地方，并与开发团队重新定义需求。
- ❑ 该过程一直持续到用户认为该原型能成功地体现想象中的系统的主要部分功能为止。
- ❑ 在这期间，用户和开发团队都不要为程序算法或设计技巧等枝节问题分心，而是要确定开发团队是否理解了用户的意思，同时试验实现它们的若干方法。
- ❑ 有了满意的系统原型，同时也积累了使用原型的经验，用户常会提出新目标，从而进一步重新原型周期。新目标的范围要比修改或补充不满意的原型大。





软件原型是软件的最初版本，以最少的费用、最短的时间开发出的、反映最后软件的主要特征的系统。它具有以下特征：

- ❑ 它是一个可实际运行的系统。
- ❑ 它没有固定的生存期。一种极端是扔掉原型（以最简便方式大量借用已有软件，做出最后产品的模型，证实产品设想是成功的，但在产品中并不使用这些模块）；另一种极端是最终产品的一部分即增量原型（先做出最终产品的核心部分，逐步增加补充模块），演进原型居于其中（每一版本扔掉一点，增加一点，逐步完善至最终产品）。
- ❑ 从需求分析到最终产品都可作原型，即可为不同目标作原型。
- ❑ 它必须快速、廉价。
- ❑ 它是迭代过程的集成部分，即每次经用户评价后修改、运行，不断重复双方认可。





原型（迭代）模型的优点：

- ❑ 任何功能一经开发就能进入测试以便验证是否符合产品需求。
- ❑ 帮助导引出高质量的产品要求。如果没有可能在一开始就弄清楚所有的产品需求，它们可以分批取得。而对于已提出的产品需求，则可根据对现阶段原型的试用而做出修改。
- ❑ 风险管理可以在早期就获得项目进程数据，可据此对后续的开发循环做出比较切实的估算。提供机会去采取早期预防措施，增加项目成功的机率。
- ❑ 大大有助于早期建立产品开发的配置管理，产品构建（build），自动化测试，缺陷跟踪，文档管理。均衡整个开发过程的负荷。
- ❑ 开发中的经验教训能反馈应用于本产品的下一个循环过程，大大提高质量与效率。
- ❑ 如果风险管理发现资金或时间已超出可承受的程度，则可以决定调整后续的开发，或在一个适当的时刻结束开发，但仍然有一个具有部分功能的，可工作的产品。
- ❑ 心理上，开发人员早日见到产品的雏型，是一种鼓舞。
- ❑ 使用户可以在新的一批功能开发测试后，立即参加验证，以便提供非常有价值的反馈。
- ❑ 可使销售工作有可能提前进行，因为可以在产品开发的中后期取得包含了主要功能的产品原型去向客户作展示和试用。





作为产品开发技术时的主要缺点：

- ❑ 如果所有的产品需求在一开始并不完全弄清楚的话，会给总体设计带来困难及削弱产品设计的完整性，并因而影响产品性能的优化及产品的可维护性。
- ❑ 如果缺乏严格的过程管理的话，这个生命周期模型很可能退化成为一种原始的无计划的“试一错一改”的循环模式。
- ❑ 心理上，可能产生一种影响尽最大努力的想法，认为虽然不能完成全部功能，但还是造出了一个有部分功能的产品。
- ❑ 如果不加控制地让用户接触开发中尚未测试稳定的功能，可能对开发人员及用户都产生负面的影响。

作为需求分析技术时的主要缺点：

- ❑ “模型效应”或“管中窥豹”。对于开发者不熟悉的领域把次要部分当作主要框架，做出不切题的原型。
- ❑ 原型迭代不收敛于开发者预先的目标。即每次更改，为了消除错误，次要部分越来越大，“淹没”了主要部分。
- ❑ 原型过快收敛于某些特定需求，而忽略了一些基本点，特别是与系统基本性能有关的部分。
- ❑ 资源规划和管理较为困难，随时更新文档也带来麻烦、版本控制可能导致失控。
- ❑ 长期在原型环境上开发，只注意得到满意的原型，容易“遗忘”用户环境和原型环境的差异。





原型开发可以应用于软件生存周期的不同阶段，也可以替代生存期的部分或全部阶段，具体可以用于以下领域：

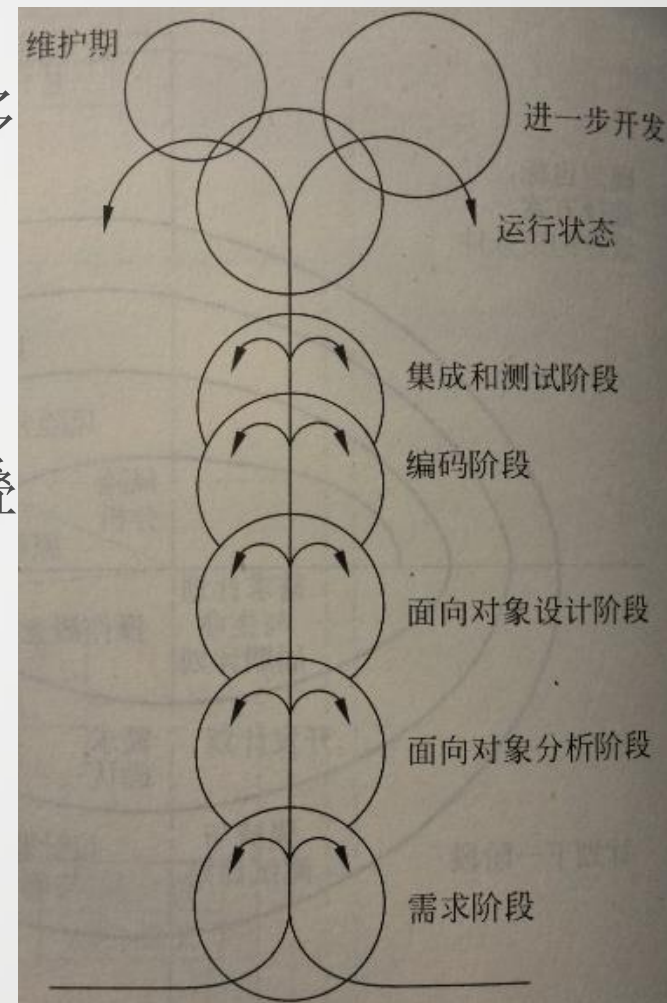
- ❑ 辅助分析和确定用户需求的任务。
- ❑ 作为软件设计的一种工具。例如：研究系统设计的可行性和适应性。
- ❑ 作为一种解决不确定性的工具。例如：研究一种新技术的效果，逐步使其适应预定的环境。
- ❑ 作为一种实验工具。
- ❑ 充作同步培训工具。
- ❑ “一次性”的验证应用。例如写一个能运行的程序，一旦得到答案，该程序将不再使用。
- ❑ 作为软件开发的辅助工具。特别是在用户需求不稳定，维护工作量很大的情况下，要求大量的重新设计工作。





基本思想:

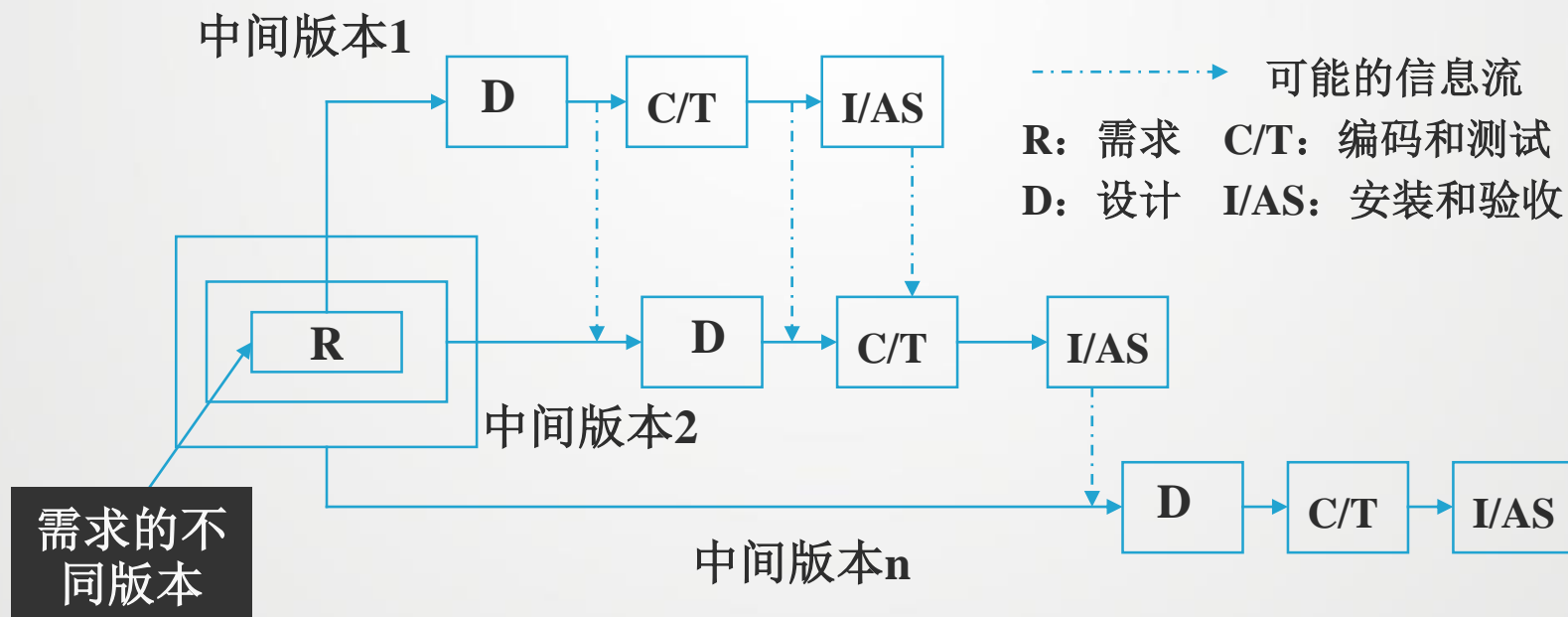
- ❑ 喷泉模型又称快速应用开发模型（Rapid Application Development Model, RAD）
- ❑ 喷泉模型认为软件生命周期的各个阶段是相互重叠和多次反复的。
- ❑ 也是一种线性开发模型
- ❑ 与瀑布模型类似，只是从串行改并行。
- ❑ 主要用于面向对象方法中，面向对象的分析和设计重叠，交叉、并行进行。





基本思想:

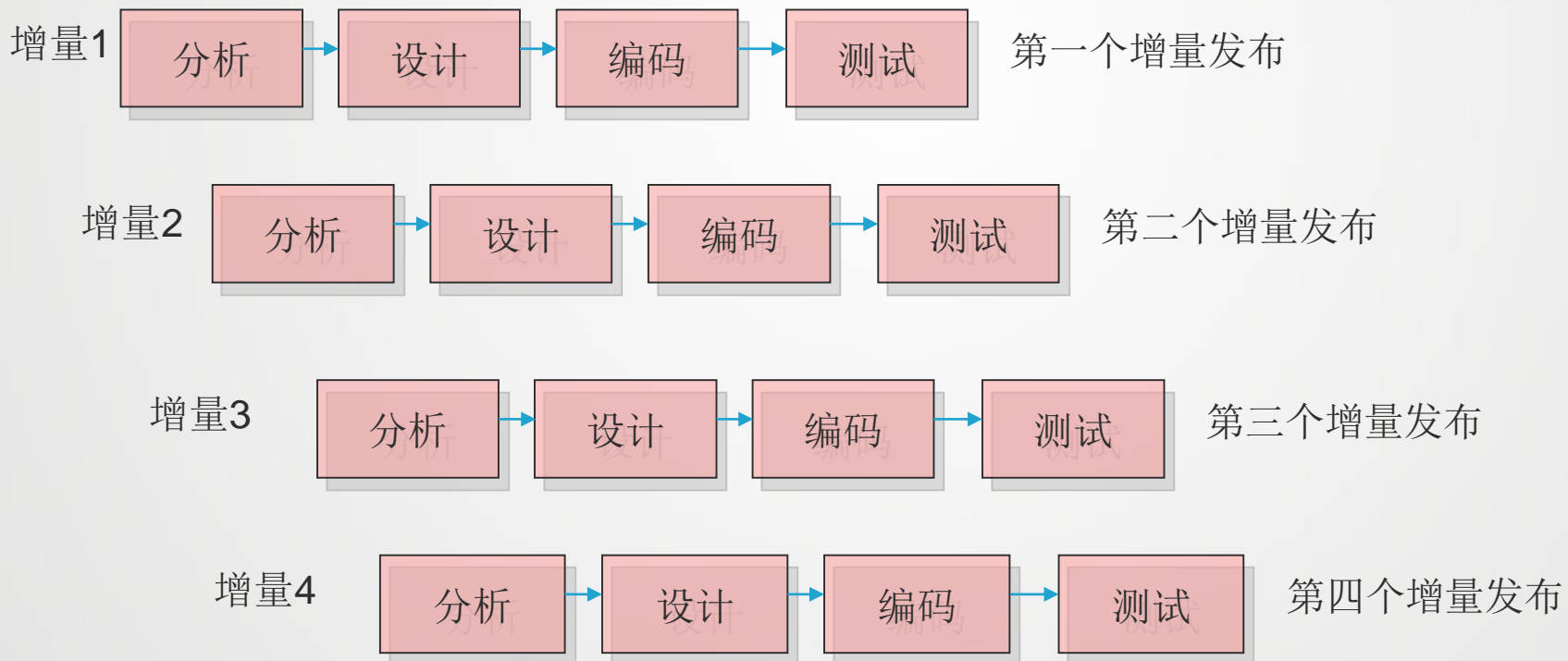
- ❑ 增量模型又称产品改进模型（Incremental Model）
- ❑ 从给定需求开始，通过构造一系列中间版本来实施开发活动，依次类推，直到系统完成。
- ❑ 每一个中间版本都是需求分析、设计、编码和测试的过程。
- ❑ 某些中间版本的开发可以并行进行。





主要特征：

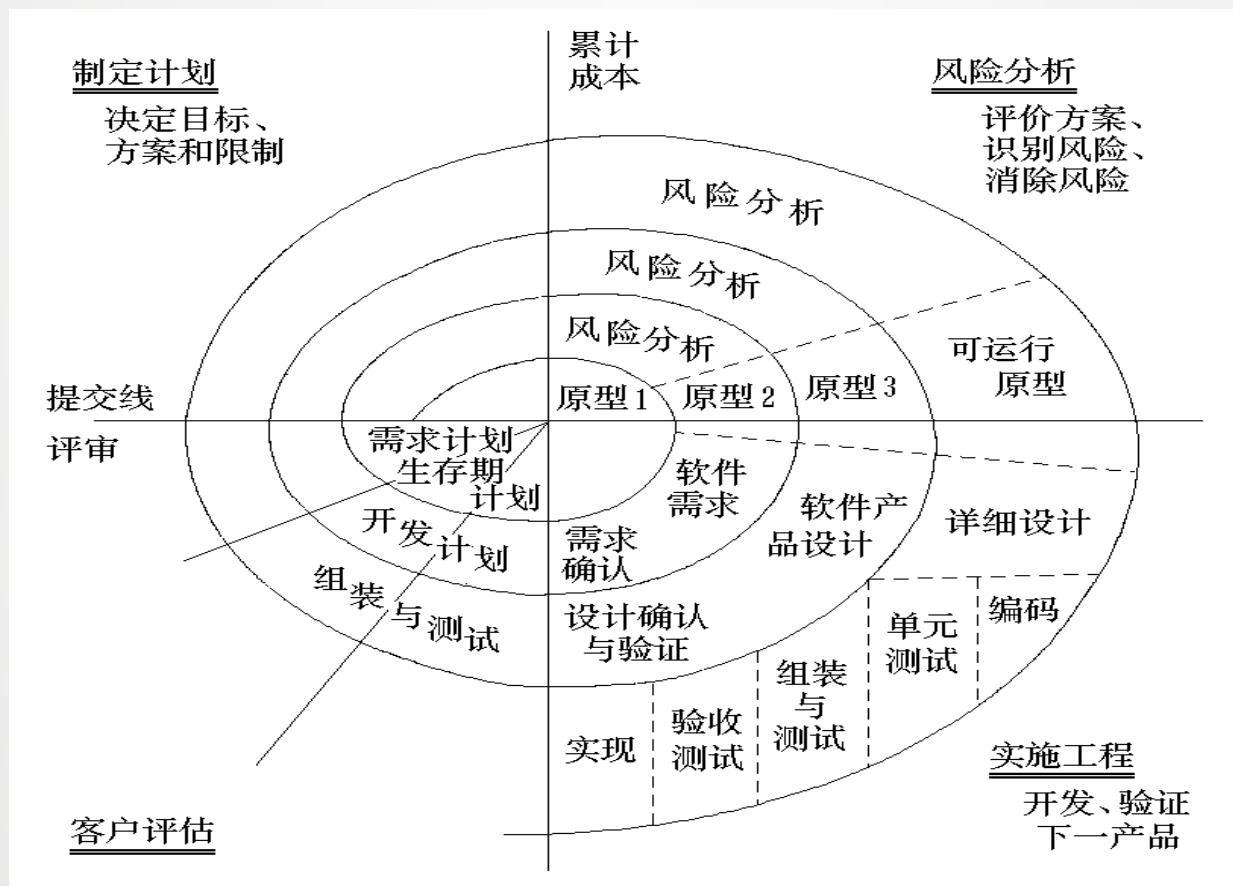
- ❑ 融合了线性顺序模型的基本成分和原型的迭代特征。
- ❑ 是随着日程时间的进展而交错的线性序列。
- ❑ 与原型不一样的地方是强调每个增量均发布一个可操作产品。
- ❑ 最典型的应用是同一个产品的不同项目（合同、用户）版本





基本思想:

- ❑ 在原型基础上，进行多次原型反复并增加风险评估，形成螺旋模型。
- ❑ 在螺旋模型结构中，维护只是螺旋模型的另一个周期，在维护 and 开发之间本质上并没有区别，从而解决了做太多测试或未作足够测试所带来的风险。





螺旋模型的优点：

- ❑ 强调严格的全过程风险管理。
- ❑ 强调各开发阶段的质量。
- ❑ 提供机会检讨项目是否有价值继续下去。

螺旋模型的缺点：

- ❑ 必须引入非常严格的风险识别，风险分析和风险控制，这对风险管理的技能水平提出了很高的要求。也需要人员，资金和时间的大量投入。



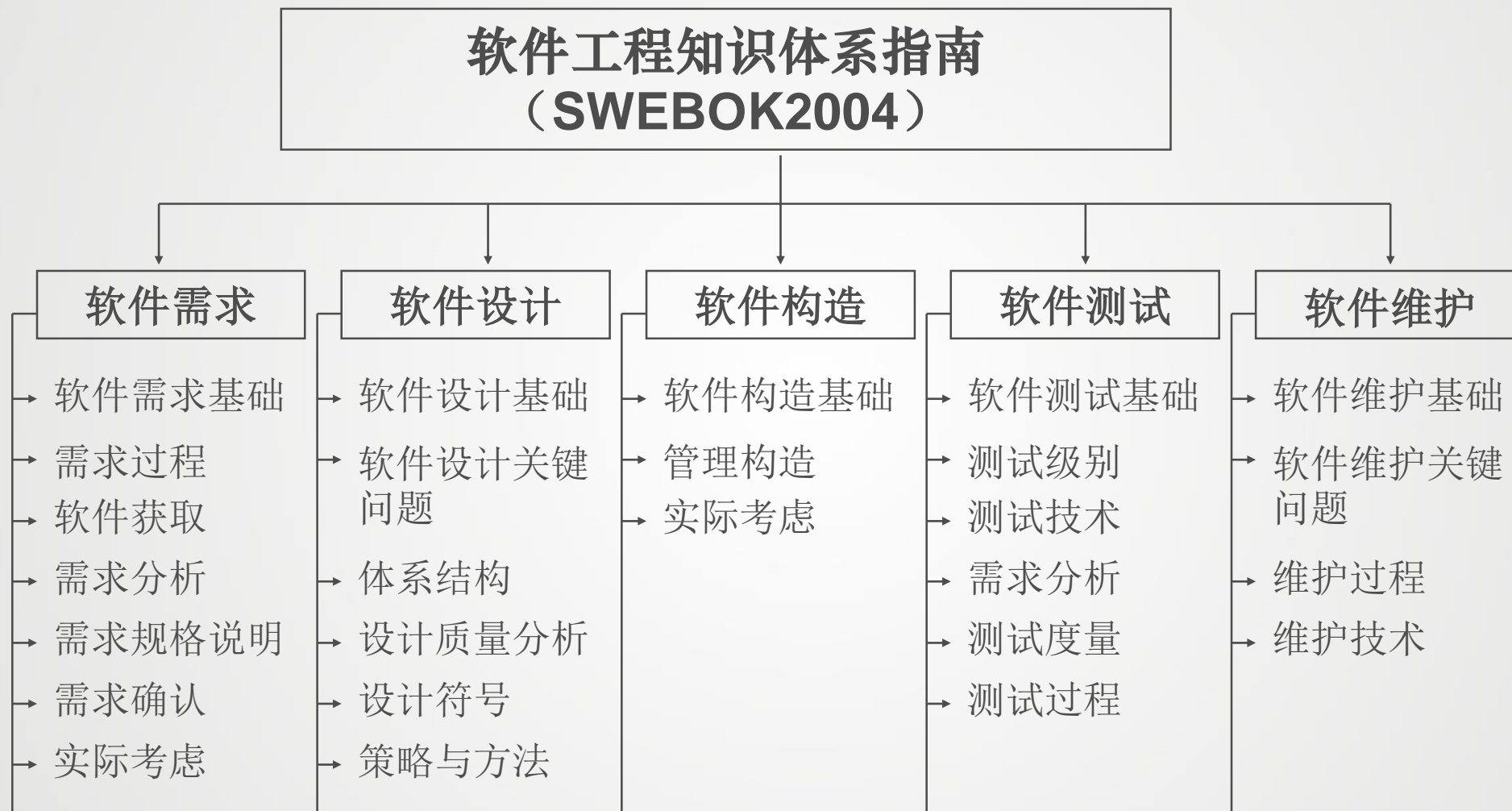


5. 软件工程知识体系SWEBOK

2004年6月，美国IEEE-CS和ACM编制的《软件工程知识体系指南2004》(SWEBOK 2004)正式发布，为软件工程知识体系本身以及相关知识之间确立了明确的边界，是我们认识软件工程学科知识体系的很好参考。

- ❑ 整个知识体系分为11个知识领域（Knowledge Area, KA）
- ❑ 前五个知识领域是按软件开发生命期阶段（按所谓瀑布模型）排列的
 - 软件需求(又称软件分析)
 - 软件设计
 - 软件构造(即编码、实现)
 - 软件测试
 - 软件维护
- ❑ 后六个知识领域则是软件开发中的支撑性或者辅助性的方面，一般覆盖软件开发的多个阶段甚至所有阶段，包括：软件配置管理(SCM)、软件项目管理、软件工程过程、软件工程工具与方法、软件质量、相关学科知识领域
- ❑ 每个知识领域之下又分为多个子领域，子领域又由多个主题构成。

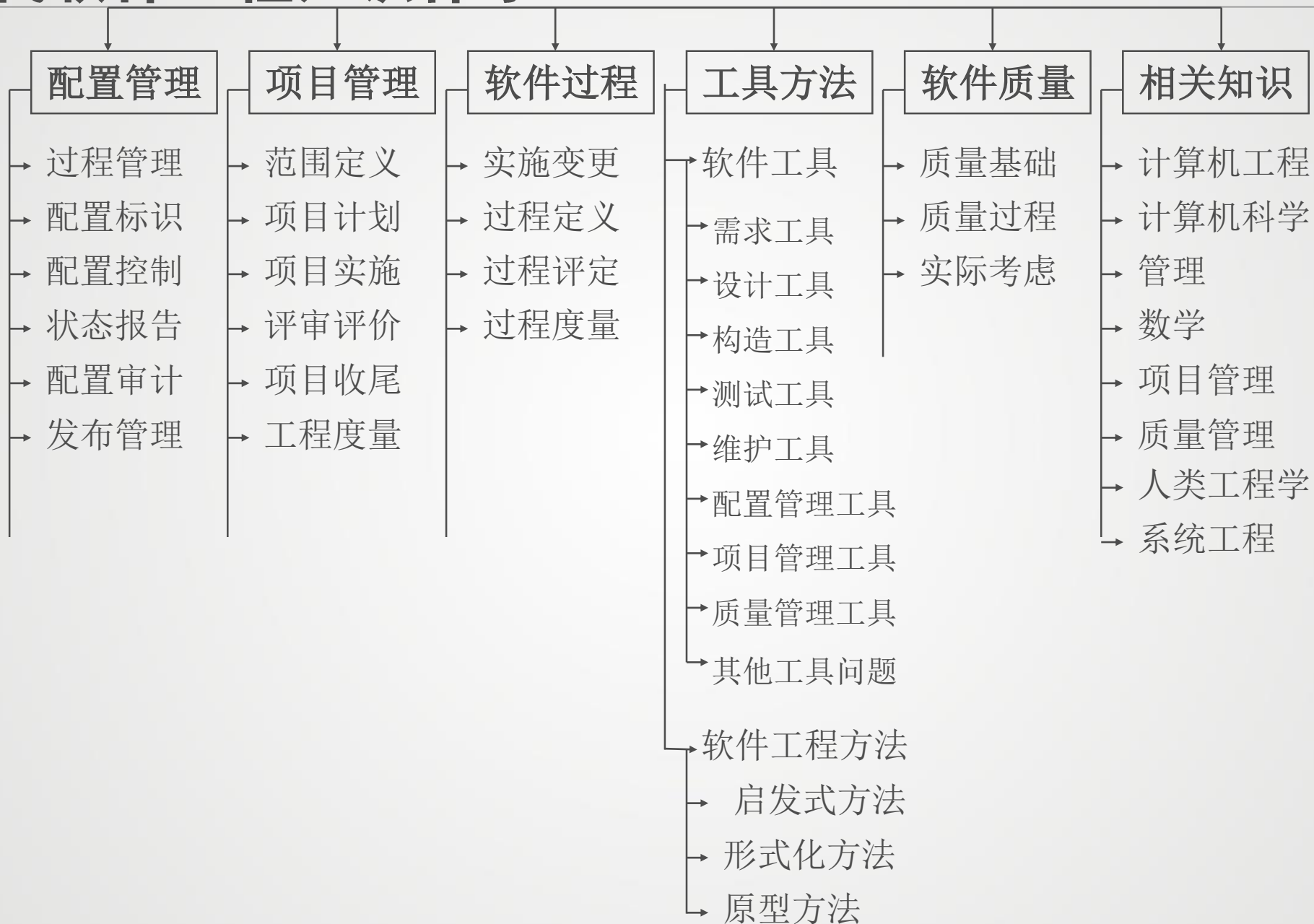


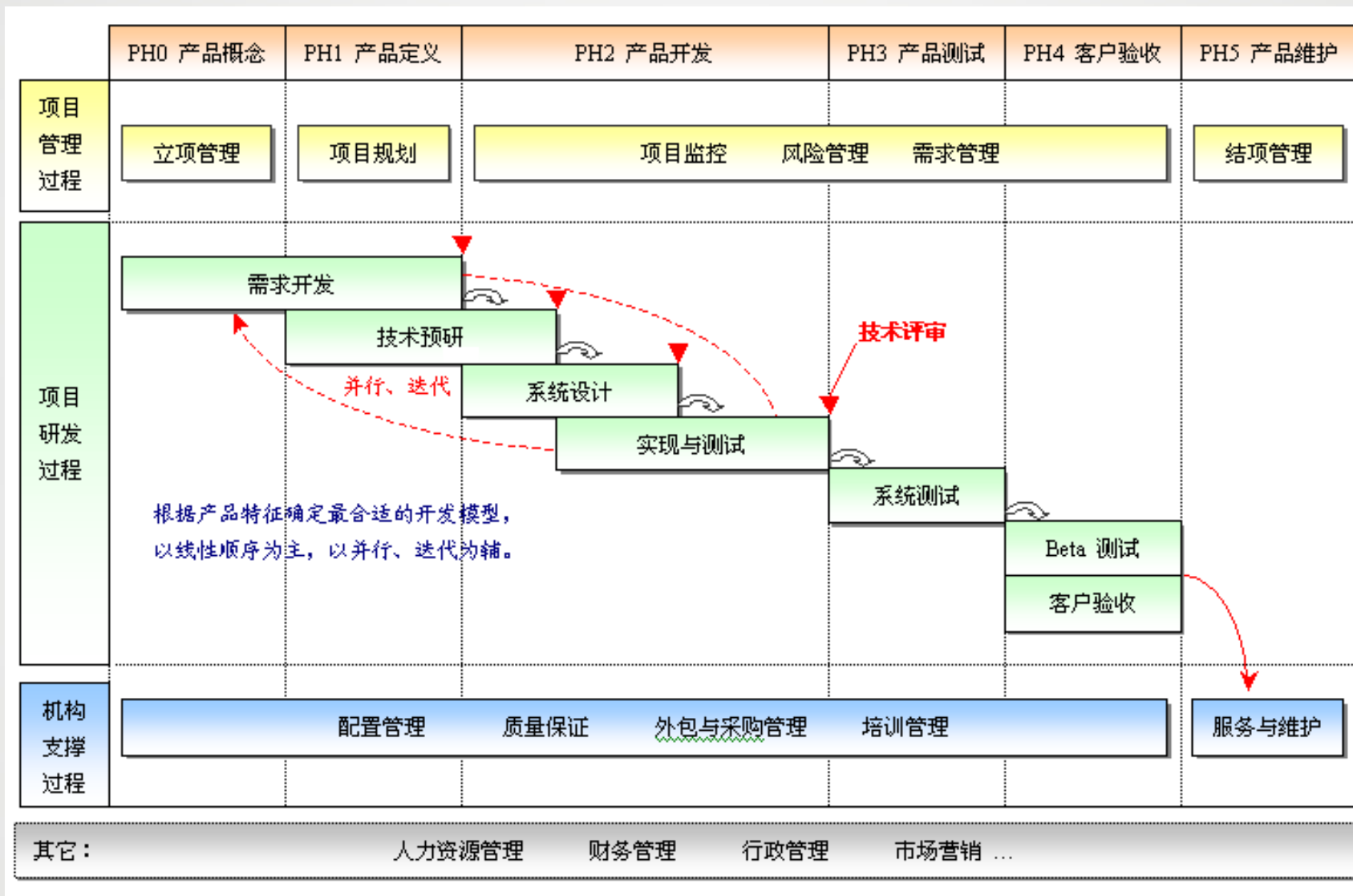




现代软件工程知识体系SWEBOK

40





6. 项目管理概述

自从有了人类，人们就开展了各种有组织的活动。随着社会的发展，有组织的活动逐步分化为两种类型：

- ❑ 一类是连续不断、周而复始的活动，人们称之为“**运作**”（**Operations**），如企业日常的产品生产活动；
 - 典型的例子：流水线的生产。
- ❑ 另一类是临时性、一次性的活动，人们称之为“**项目**”（**Projects**），如企业的一次技术改造活动、一项建设工程的实施。

在当今社会中，一切都是项目，一切也将成为项目。

——美国项目管理专业资质认证委员会主席 Paul Grace



项目是一个特殊的、将被完成的有限任务，它是在一定时间内，满足一系列特定目标的多项相关工作的总称。

理解项目的定义，实际包含三层含义：

- ❑ 项目是一项有明确目标并待完成的任务；
- ❑ 项目是在一定的环境下及在一个特定的组织机构内，利用有限资源（人力、物力、财力等）在规定的时间内（时间也是有限的资源）内完成任务；
- ❑ 任务的完成要满足一定功能、性能、质量等具体技术指标要求。





项目具有如下一些特点：

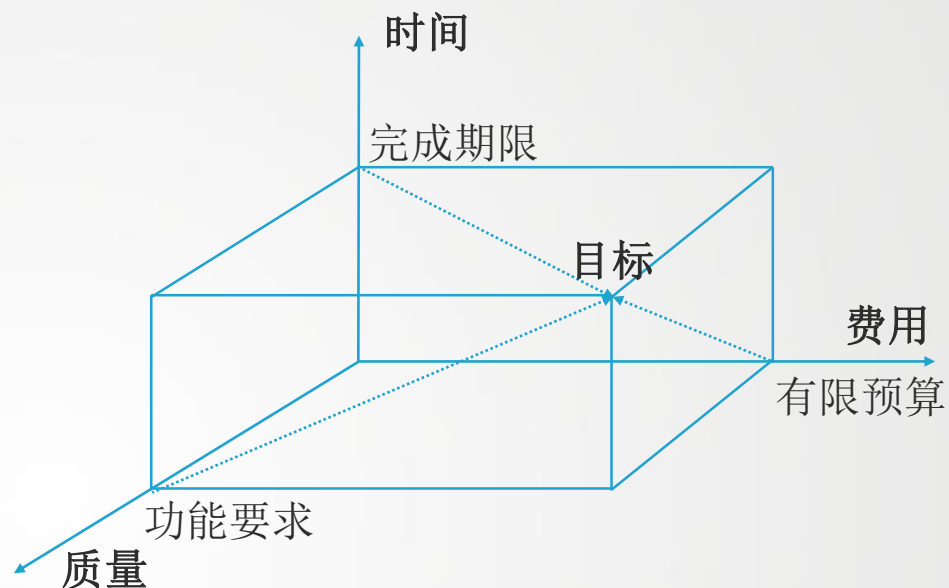
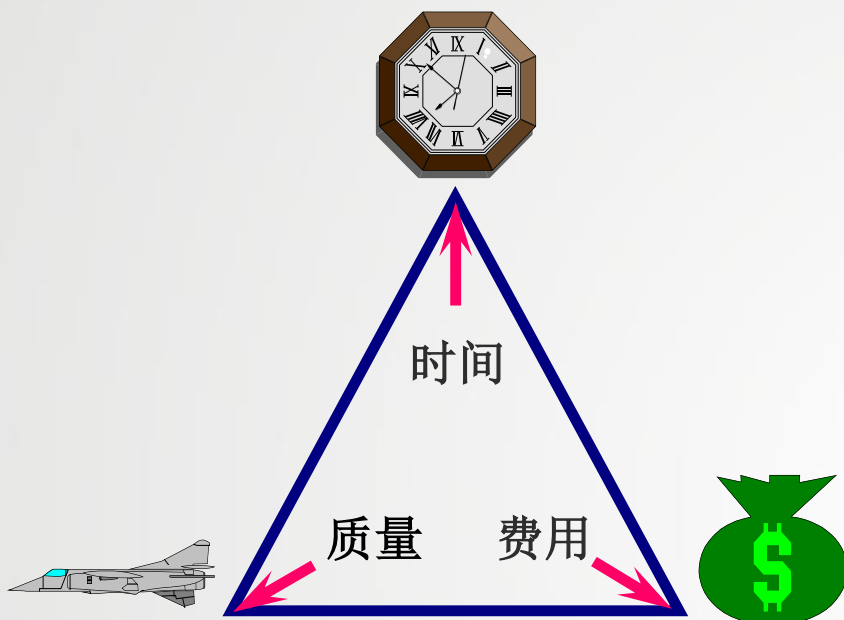
- ☐ 一次性：有明确的开始/结束时间
- ☐ 独特性：没有以完全相同方式、完全相同的人完成
- ☐ 目标确定性/过程的不确定性
- ☐ 活动的整体性、协调性/过程的渐进性
- ☐ 结果的不可逆转性
- ☐ 资源的依赖性/成本的约束性
- ☐ 团队的临时性与开发性





项目的三重约束

45



成功的项目必须满足客户、管理层和供应商在时间、费用和性能上的不同要求。





项目管理就是以项目为对象的系统管理方法，通过一个临时性的、专门的柔性组织，运用相关的知识、技术和手段，对项目进行高效率的计划、组织、指导和控制，以实现项目全过程的动态管理和项目目标实现过程的综合协调与优化。

理解项目管理的定义，实际包含四层含义：

- ❑ 项目管理的对象——项目
- ❑ 项目管理的组织特点——临时性、富有柔性
- ❑ 项目管理的手段——计划、组织、指导和控制
- ❑ 项目的管理目标——实现项目全过程的动态管理及项目目标实现过程的综合协调与优化

有效的项目管理是指在规定用来实现项目目标和指标的时间内，对组织机构资源进行计划、引导和控制工作。

——美国项目管理学会《项目管理知识体系纲要》





项目管理具有如下一些特点：

- ❑ 项目管理的**对象**是项目或被当作项目来处理的运作。
- ❑ 项目管理的**思想**是管理的系统方法论。
- ❑ 项目管理的**组织**通常是临时性、柔性、扁平化的组织。
- ❑ 项目管理的**机制**是项目经理负责制，强调责权利的对等。
- ❑ 项目管理的**方式**是目标管理，包括：进度、费用、技术与质量的综合协调优化。
- ❑ 项目管理的**要点**是创造和保持一种使项目顺利进行的氛围与环境。
- ❑ 项目管理的**方法、工具和手段**具有先进性和开放性。

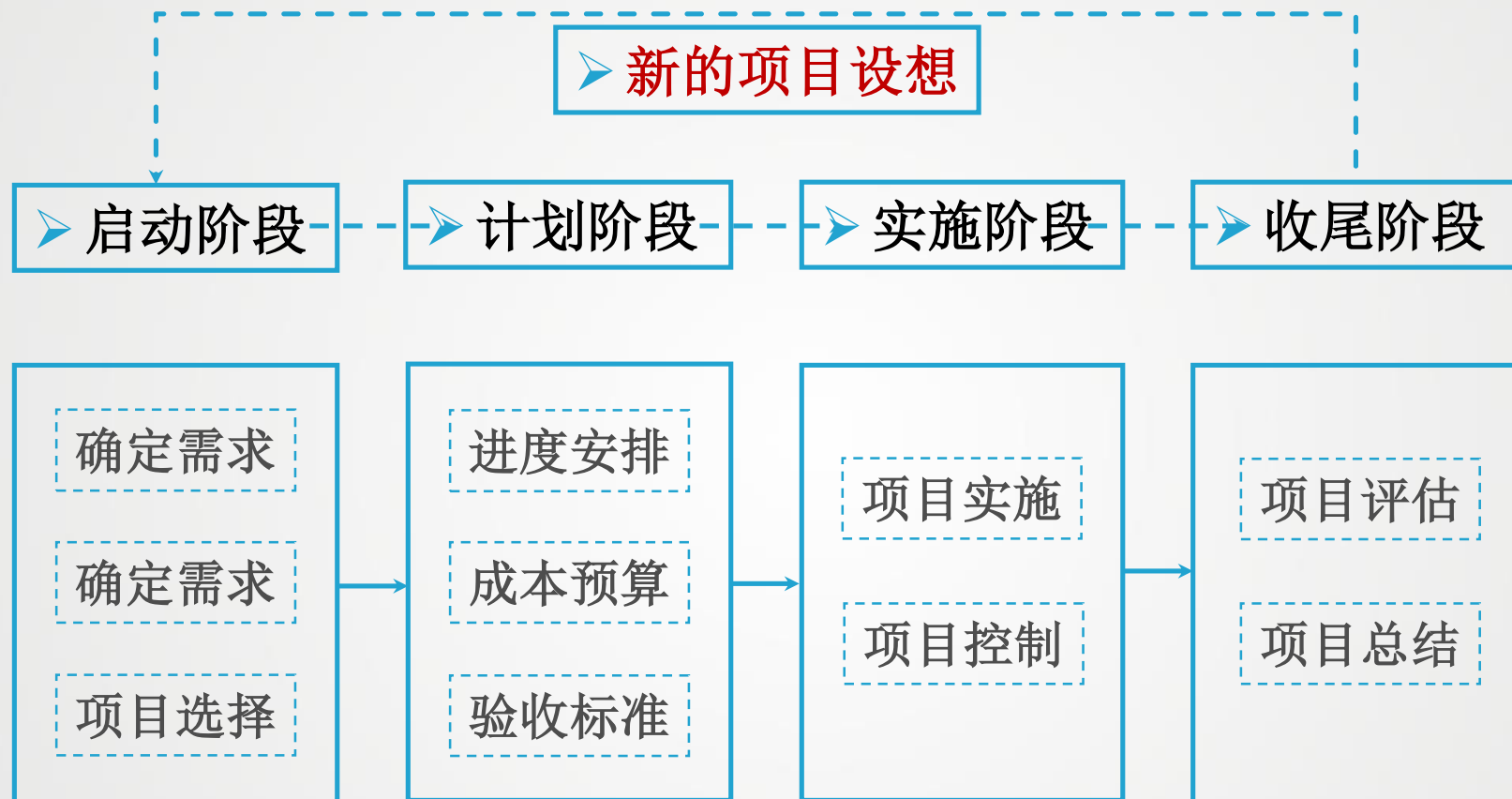




项目管理的基本要素如下：

- ☐ 项目的生命周期
- ☐ 项目干系人
- ☐ 项目资源
- ☐ 项目目标
- ☐ 用户需求







项目启动阶段的主要工作：

- ☐ 明确需求、策划项目
- ☐ 调查研究、收集数据
- ☐ 确立目标
- ☐ 进行可行性研究
- ☐ 明确合作关系
- ☐ 风险分析
- ☐ 拟订战略方案
- ☐ 进行资源测算
- ☐ 提出组建项目组方案
- ☐ 提出项目建议书
- ☐ 获准进入下一阶段





项目计划阶段的主要工作：

- ☐ 确定项目组主要成员
- ☐ 项目最终产品的范围确定
- ☐ 实施方案研究
- ☐ 项目质量标准的确定
- ☐ 项目的资源保证
- ☐ 项目的环境保证
- ☐ 主计划的制订
- ☐ 项目经费及现金流量的预算
- ☐ 项目的工作结构分解（WBS）
- ☐ 项目政策与过程的制订
- ☐ 风险评估
- ☐ 确认项目有效性
- ☐ 提出项目概要报告
- ☐ 获准进入下一阶段





项目实施阶段的主要工作：

- ☐ 建立项目组织
- ☐ 建立与完善项目联络渠道
- ☐ 实施项目激励机制
- ☐ 建立项目信息控制系统
- ☐ 建立项目工作包，细化各项技术需求
- ☐ 执行WBS的各项工
- ☐ 获得订购物品及服务
- ☐ 指导/监督/预测/控制：范围、质量、进度、成本
- ☐ 解决实施中的问题



检查点：

- ❑ 指在规定的时间内对项目进行检查，比较实际情况与计划之间的差异，并根据差异进行调整。
- ❑ 可将检查点看作是一个固定“采样”时点，而时间间隔则根据项目周期长短不同而不同，频度过小会失去意义，频度过大会增加管理成本。
- ❑ 常见的间隔是每周一次，项目经理可通过周报告、召开例会等形式，进行检查。

里程碑：

- ❑ 完成阶段性工作的标志，不同类型的项目，里程碑也不同。里程碑在项目管理中具有重要意义。

基线：

- ❑ 是一个（或一组）配置项在项目生命周期的不同时间点上通过正式评审而进入正式受控的一种状态。
- ❑ 在项目管理中，需求、进度、成本、质量等基线，都是一些重要的项目阶段里程碑，但相关交付物要通过正式评审并作为后续工作的基准和出发点——即达到基线标准。
- ❑ 基线一旦建立后变化需要受控制。





总结：

- ❑ 项目生命周期可以分成项目启动、计划、实施和收尾四个阶段。
- ❑ 项目应该在检查点进行检查，比较实际和计划的差异并进行调整；
- ❑ 通过设定里程碑，来增强控制、降低风险；而基线是重要的里程碑，交付物应通过评审并开始受控。



项目参与人：

- ❑ 项目参与人是指项目的直接参与各方。
- ❑ 建筑工程的参与方有：建设方、投资方、贷款方、承包人、供货商、建筑/设计师、监理工程师、咨询顾问等。
- ❑ 项目参与人：甲方、乙方，第三方：监理

项目干系人：

- ❑ 项目干系人的概念，比参与人要广泛的多
- ❑ 项目干系人包括项目参与人和其利益受该项目影响（受益或受损）的个人和组织；也可以把他们称作项目的利害关系者。
- ❑ 除了项目参与外，项目干系人还可能包括政府的有关部门、社区公众、项目用户、新闻媒体、市场中潜在的竞争对手和合作伙伴等；甚至项目班子成员的家属也应视为项目干系人。





资源：

- ❑ 由于项目固有的一次性，项目资源不同于其他生产活动的资源，它多是临时拥有和使用的。
- ❑ 资金需要筹集，服务和咨询力量可采购（如招标发包）或招聘，有些资源还可以租赁。
- ❑ 项目过程中资源需求变化甚大，获得的时间和资源质量影响项目。有些资源用毕后要及时偿还或遣散，任何资源积压、滞留或短缺都会给项目增加成本。
- ❑ 资源的合理、高效的使用对项目管理尤为重要。



目标：

- ❑ 项目要求达到的目标可分为两类，必须满足的**规定要求**和隐含或附加的**期望要求**。
- ❑ **规定要求**包括：项目实施范围、质量要求、利润或成本目标、时间目标以及必须满足的法规要求等。这里指的是狭义的质量，如项目及项目成果的技术指标和性能指标等；
- ❑ 在一定范围内，质量、成本、进度三者是互相制约的；
- ❑ **期望要求**常常包括对开辟市场、争取支持、减少阻力产生重要影响。譬如：一种新产品，除了基本性能之外，外形、色彩、使用舒适，建设和生产过程有利于环境保护和改善等，也应当列入项目的目标之内。

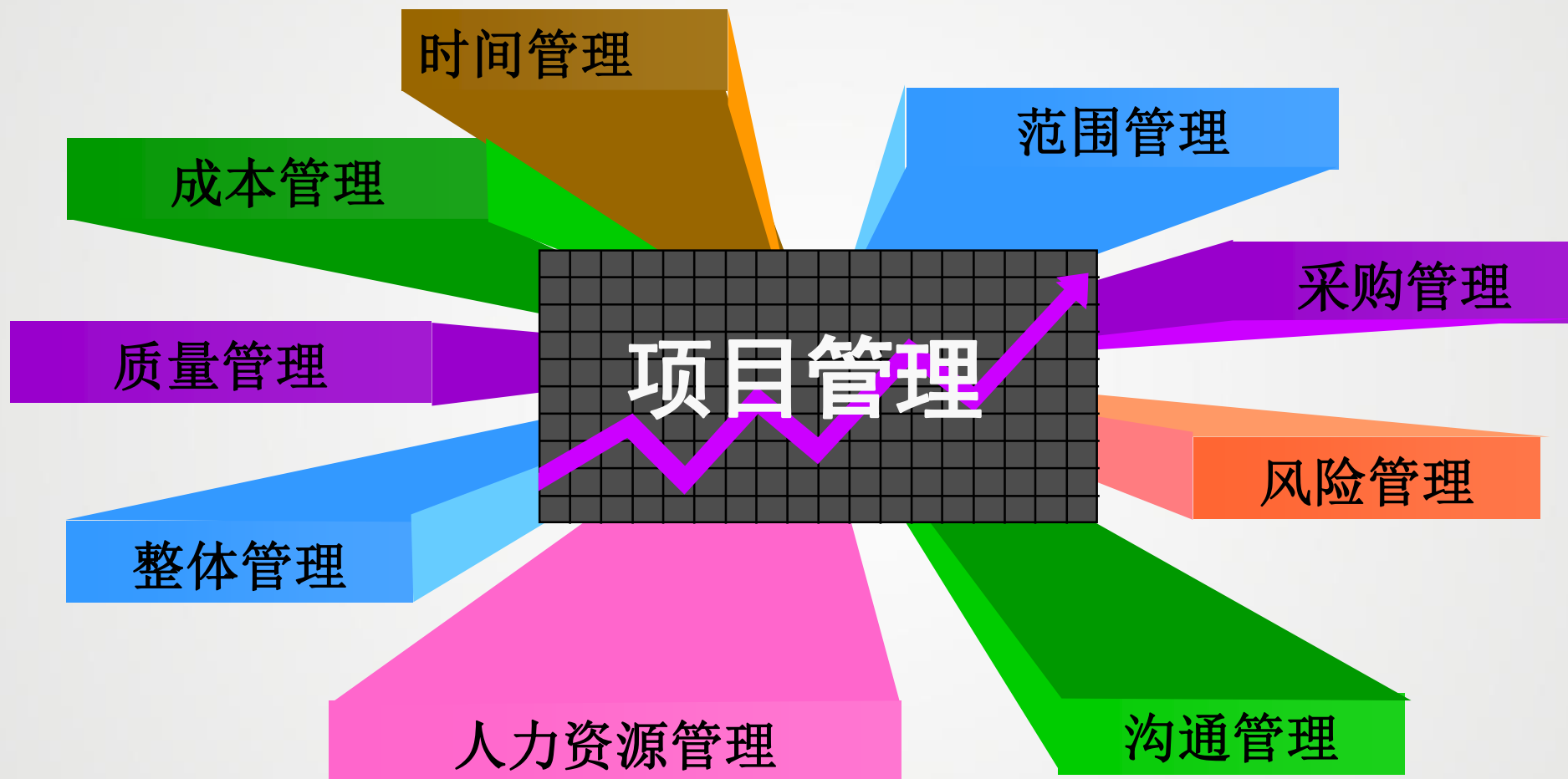




需求:

- ❑ 项目要求达到的目标是根据需求和可能来确定的。
- ❑ 一个项目的各种不同干系人有各种不同的需求，有的相去甚远，甚至互相抵触。这就要求项目管理者对这些不同的需求加以协调，统筹兼顾，以取得某种平衡，最大限度地调动项目干系人的积极性，减少他们的阻力和消极影响。
- ❑ 项目干系人的需求往往是笼统的、含糊的，他们有时缺乏专门知识，难以将其需求确切、清晰地表达出来。因此需要项目管理人员与干系人充分合作，采取一定的步骤和方法将其确定下来，成为项目要求达到的目标。
- ❑ 项目干系人在提出其需求时，未必充分地考虑了其实现的可能性。项目管理者还应协助用户进行可行性研究，评估项目的得失，调整项目的需求，优化项目的目标。有时可引导用户和其它干系人去追求进一步的需求，有时要帮助他们放弃不切实际的需求，有时甚至要否定一个项目，避免不必要的损失。
- ❑ 项目干系人的需求在项目进展过程中往往还会发生变化，项目需求的变化将引起项目目标、范围、计划等一系列相应的变化。因此，根据需求进行范围管理自始至终都是项目管理中极为重要的内容。
- ❑ 在软件项目中，对需求的控制和管理，发展为软件需求工程。









项目管理组织是按照项目的目标以一定的形式组建起来的，由组织各部门调集专业人才，并指派项目负责人在特定时间内完成任务。

项目管理组织结构的基本形式：

- ☐ 项目型组织结构
- ☐ 职能型组织结构
- ☐ 矩阵型组织结构





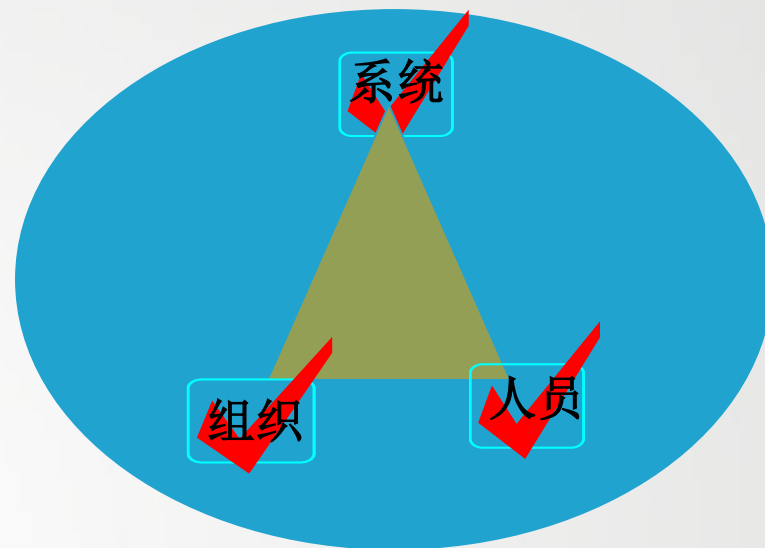
- ☐ 目标的一致性和管理的统一
- ☐ 有效的管理幅度和层次
- ☐ 责任和权利对等
- ☐ 合理分工和密切协作
- ☐ 集权与分权相结合
- ☐ 纪律和秩序
- ☐ 团队精神





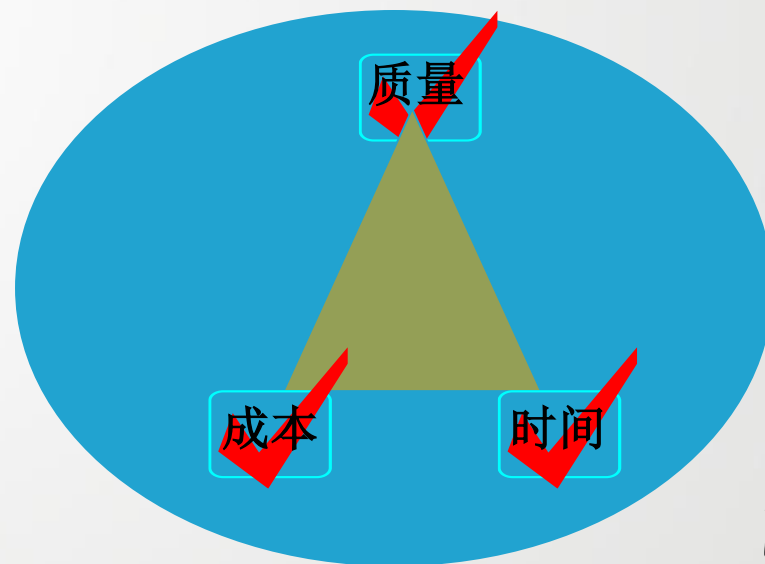
大三角：

- 通过一系列的领导及管理活动，使项目的目标成功实现，并使项目干系人都获得满意



小三角：

- 控制项目的实施在预定的质量、成本和时间范围内





对于所属组织的责任:

- ❑ 保证项目的目标符合于组织目标
- ❑ 充分利用和保管组织分配给项目的资源
- ❑ 及时与组织高层就项目进展进行沟通





对于所管项目的责任:

- ❑ 明确项目目标及约束
- ❑ 制定项目的各种活动计划
- ❑ 确定适合于项目的组织机构
- ❑ 招募项目组成员，建设项目团队
- ❑ 获取项目所需资源
- ❑ 领导项目团队执行项目计划
- ❑ 跟踪项目进展及时对项目进行控制
- ❑ 处理与项目相关者的各种关系
- ❑ 项目考评与项目报告





- ❑ (1) 密切保持与用户和本组织上层领导的联系，及时沟通项目合同执行中的重要信息。
- ❑ (2) 熟悉合同，了解本组织和用户的意图和情况，制定项目计划和项目协调程序，确定项目实施的基本工作方法和程序，经用户和公司批准后执行。
- ❑ (3) 代表开发商，参加与用户的协调会议。
- ❑ (4) 根据项目任务范围，确定项目实施组织，落实项目团队的成员，评价项目团队主要人员能否胜任项目工作。
- ❑ (5) 提出项目的工作分解结构（WBS）和组织分解结构（OBS），并确定其编码系统。
- ❑ (6) 组织项目内部会议，制定项目的工作任务、范围以及工程进度/费用控制计划。
- ❑ (7) 协调项目实施过程中的工作关系，包括对外与用户及其他协作单位之间的关系，对内与各有关专业职能部、室的关系。
- ❑ (8) 委派专人负责文件管理，以保证项目函电、会议纪要、备忘录、图纸资料及时处理，并保证工程档案的完整性。
- ❑ (9) 审核项目的质量计划、财务计划、设计计划、采购计划、施工计划。
- ❑ (10) 审查项目的进度计划、费用估算和预算。





- ❑ **(11)** 从合同目标的角度，审查设计方案。
- ❑ **(12)** 审查关键设备和特殊材料的采购活动。
- ❑ **(13)** 审查和分析项目进展报告，预测项目实施中可能出现的问题，并提出预防措施及解决办法。重大问题应及时向公司领导和管理部门报告。
- ❑ **(14)** 及时处理项目变更和用户变更，同时协调可能涉及的每一个有关方面，必要时做出相应的修改和安排。
- ❑ **(15)** 与项目质量经理共同监督保证工程质量，参加质量问题的研究和处理。
- ❑ **(16)** 出现有违约事件时，及时参加谈判和参与问题的处理。对于超出合同条款规定的问题的处理，都要经过协商谈判，并写出会议纪要。
- ❑ **(17)** 组织编制项目验收申请报告，向用户提出验收申请，协调用户验收，并取得用户对项目验收的正式文件。
- ❑ **(18)** 督促检查项目资料的整理入库工作。
- ❑ **(19)** 负责审查项目结算，处理与用户及分包单位的遗留问题。
- ❑ **(20)** 组织项目的工作总结和回访工作



本章结束

***ANY
QUESTION***

