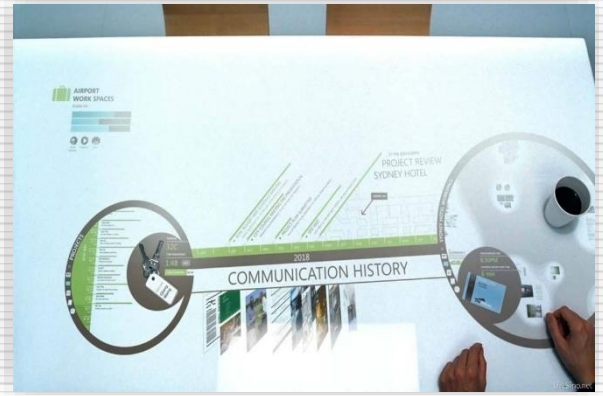
CHAPTER 1

Introduction to Software Engineering

Outline

- What is software
- History of software
- Software crisis
- Software engineering
- Software life cycle
- Process model of software development

1.1 What is software



软件无处不在，为我们的生活创造了无限精彩。在当今的信息时代，世界正在变得更加“智慧”，万事万物间感知化、互联化和智能化的程度不断加深。

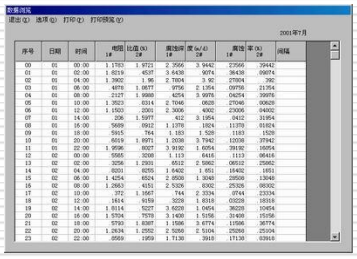
软件工程为这一切做出了巨大贡献，随着该学科的成熟发展，其未来的贡献将不可限量。

Software defines world

Software defines everything

What is software

Software:



软件

软件是计算机系统中与硬件相互依存的另一部分，它是包括程序、数据及其相关文档组成的完整集合。

软件=程序+数据+文档。

程序：程序是按事先设计好的功能和性能要求执行的指令序列。

数据：数据是指程序能正常处理信息的数据和数据结构。

文档：文档是与软件开发、设计、运行、维护有关的图文资料。

Program

Program is the ordered instruction sequences executed by computer in order to run special task.

- 程序的形式

Machine oriented program **= binary codes**

Procedure oriented program **= algorithms + data structures**

Object oriented program **= objects + messages**

Component oriented program **= components + infrastructure**

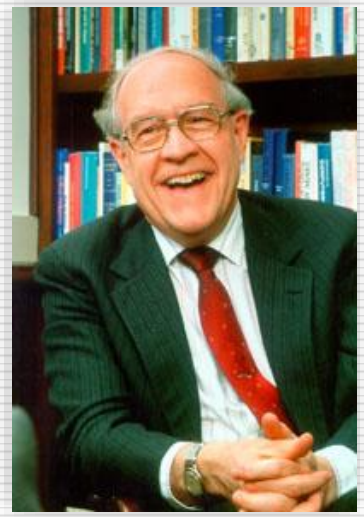
Document Usages（文档的作用）

- ✓memory, record
- ✓communication tool
- ✓milestone
- ✓base of management
- ✓step of quality guarantee
- ✓training and reference
- ✓support in maintenance

软件的特点

- ✓ 软件看不见摸不着；
- ✓ 软件是一种逻辑实体，而不是具体的物理实体。因而它具有抽象性；
- ✓ 软件的生产与硬件不同，在它的开发过程中没有明显的制造过程；
- ✓ 在软件的运行和使用期间，没有硬件那样的机械磨损，老化问题；
- ✓ 软件的开发和运行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的开发方式；
- ✓ 软件成本相当昂贵。

软件的本质特性



软件具有**复杂度**、**一致性**、**可变性（演化性）**和**不可见性**等固有的内在特性，这是造成软件开发困难的根本原因。

Brooks, F. P., “No silver bullet: essence and accidents of software engineering”, *IEEE Computer* , Vol. 20, No. 4, pp.10-19, 1987

1.2 Classification of software

- ✓ **function**: system software, eg os; support software, eg DB, CASE; application software
- ✓ **size:** small-scale, medium-scale, large-scale
- ✓ **work mode**: real-time, time-share, interactive, batch
- ✓ **service**: project, product
- ✓ **sale:** contract, ordered, non-contract

软件分类

✓系统软件：系统软件是为其它软件服务的软件。

✓实时软件：管理、分析、控制现实世界中所发生的事件的软件称为实时软件。

✓商业管理软件：商业信息处理是最大的软件应用领域，包括常规的数据处理软件和一些交互式的计算处理(如POS软件)软件。各类管理信息系统(MIS)、企业资源计划(ERP)、客户关系管理(CRM)等都是典型的商业管理软件。

✓工程与科学计算软件：此类软件的特征是要实现特定的“数值分析”算法。

✓嵌入式软件：驻留在专用智能产品中，用于控制这些产品进行正常工作，完成很有限、很专业的功能的软件。例如各类智能检测仪表、数码相机、移动电话、微波炉等智能产品都必须在嵌入式软件的支持下才能正常工作。

软件分类

✓人工智能软件：利用非数值算法去解决复杂问题的软件。各类专家系统、模式识别软件、人工神经网络软件都属于人工智能软件。

✓个人计算机软件：文字处理系统、电子表格、游戏娱乐软件等等。

此外，还可以根据软件的规模、软件的工作方式、使用频度、失效后造成的影响等对软件产品进行分类。

软件产品分类

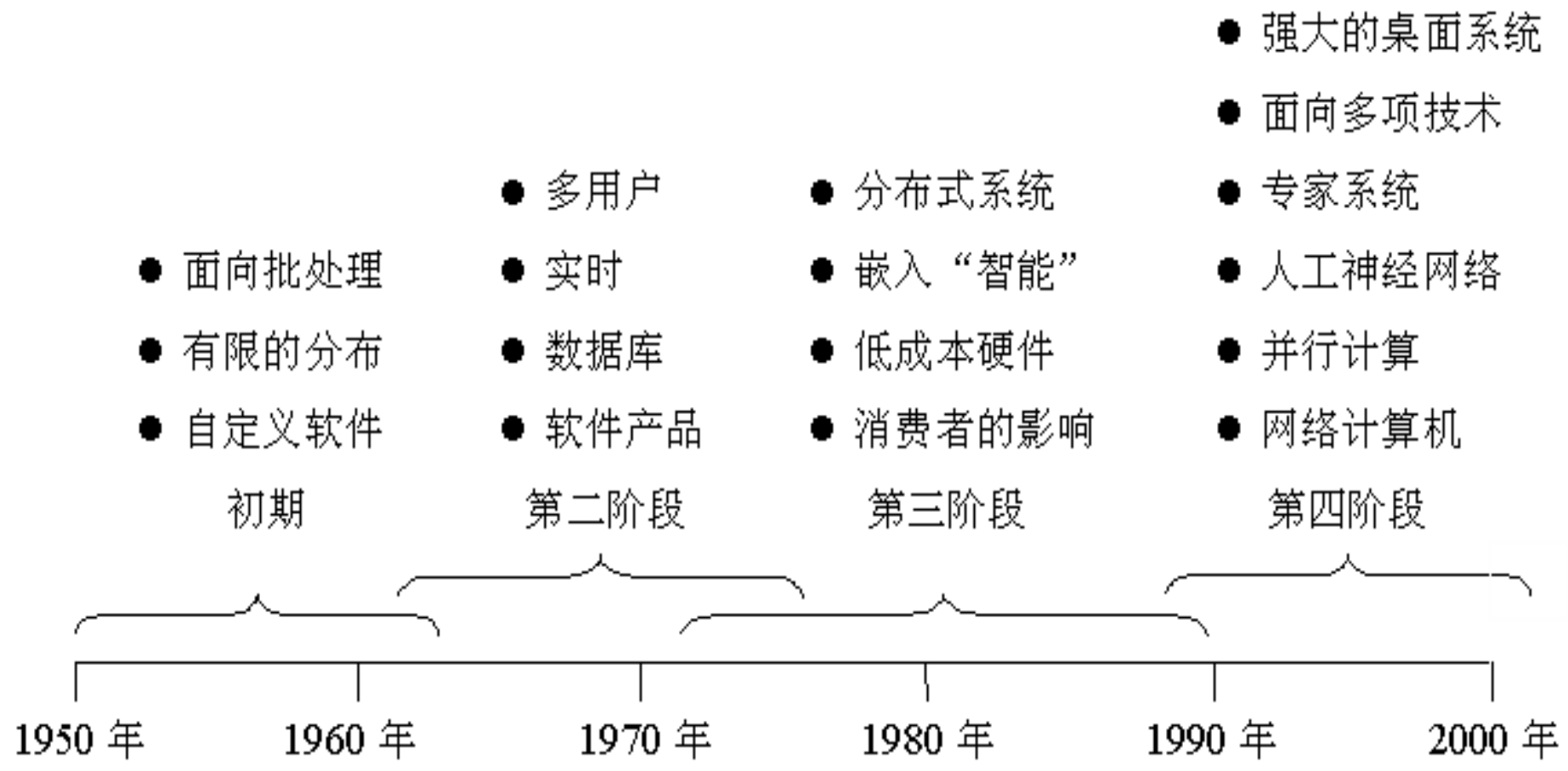
规模类别	参加人员数	开发期限	产品规模(源代码行数)
微型	1	1～4周	0.5 k
小型	1	1～6月	1～2 k
中型	2～5	1～2年	5～50 k
大型	5～20	2～3年	50～100 k
甚大型	100～1000	4～5年	1 M
极大型	2000～5000	5～10年	1～10 M

1.3 History of Software

Evolution (形态上）：

- Code
- Program
- Software
- Software Project
- Software Product

1.3 软件的发展史


面向批处理
有限的分布
自定义软件
初期
多用户
实时
数据库
软件产品
第二阶段
分布式系统
嵌入“智能”
低成本硬件
消费者的影响
第三阶段
强大的桌面系统
面向多项技术
专家系统
人工神经网络
并行计算
网络计算机
第四阶段
1950 年
1960 年
1970 年
1980 年
1990 年
2000 年

1.3 软件的发展史

1949—1959

独立编程服务

- 代表：CSC
- 特点：为单个客户提供定制软件，包括技术咨询、软件编程和维护；软件销售是一次性的。

1959—1969

软件产品

- 代表：Informatics
- 特点：保护知识产权，以许可证的方式销售软件产品。

1969—1981

企业解决方案

- 代表：Oracle
- 特点：面向领域，以企业解决方案供应商面目出现。

1981—1994

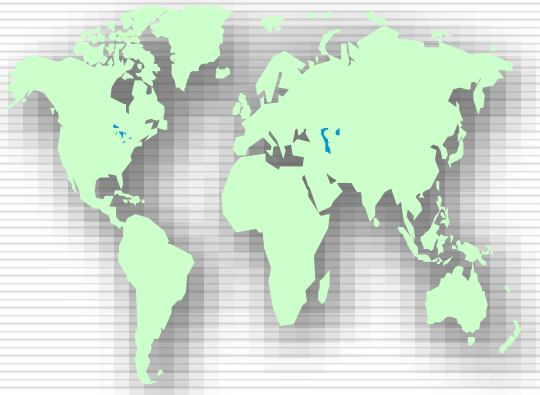
面向大众的成套软件

- 代表：Microsoft
- 特点：基于个人计算，服务大众市场。

1994—今

网络增值服务

- 代表：Google
- 特点：向端用户收小头，向寄生在服务上的广告商要大头。



软件的发展史

第一阶段: 20世纪50年代初期至20世纪60年代初期的十余年

- ✓计算机系统开发的初期阶段，尝试和体验。
- ✓编写者和使用者往往是同一个或同一组人。
- ✓隐含过程，最后除了程序清单外，没有其他文档资料保存下来。

第二阶段: 跨越了从60年代中期到70年代末期的十余年，

- ✓出现了软件产品和“软件作坊”的概念，设计人员开发软件不再像早期阶段那样只为自己的研究工作需要，而是为了用户更好地使用计算机，但“软件作坊”仍然沿用早期形成的个体式的软件开发方法。
- ✓软件的维护工作以惊人的比例耗费资源，更严重的是，程序设计的个体化和作坊化特性使软件最终成为不可维护的，从而出现了早期的软件危机。

软件的发展史

第三阶段：20世纪70年代中期至20世纪80年代末期，分布式系统、网络的发展对软件开发提出了更高的要求，

- ✓硬件的发展速度已经超过了人们对软件的需求速度，因此使得硬件价格下降，软件的价格急剧上升，导致了软件危机的加剧，致使更多的科学家着手研究软件工程学的科学理论、方法和时限等一系列问题。
- ✓软件开发技术的度量问题受到重视，最著名的有软件工作量估计COCOMO模型、软件过程改进模型CMM等。

软件的发展史

第四阶段： 从20世纪80年代末期开始的。这个阶段是强大的桌面系统和计算机网络迅速发展的时期

- ✓计算机体系结构由中央主机控制方式变为客户机/服务器方式，专家系统和人工智能软件终于走出实验室进入了实际应用，虚拟现实和多媒体系统改变了与最终用户的通信方式，出现了并行计算和网络计算的研究，面向对象技术在许多领域迅速取代了传统软件开发方法。

软件的发展现状

(1) 已经存在大量正在运行的软件。 政治、军事、金融、电信、航空航天等

(2) 软件的应用范围不断扩大。商务、交通、家电等，软件无处不在。

(3) 软件的规模与复杂性持续增加

非常大规模系统：从50万行增加到1000万行，扩大了20倍；

复杂性：a.子系统数目越来越多； b.计算机应用从数值计算开始发展到几百万条指令的大型企业业务应用，再发展到几千万终端用户直接交互工作的网络应用。

(4) 出现了大量与软件相关的标准。CORBA、UML、XML、TMN、CWM等。

(5) 软件危机存在（软件脱节）

1968-2000：软件效率、质量、进度、预算无法控制。

新世纪软件产业的趋势

- **网络化趋势**：计算机与通信的融合趋势

  万维网→智能网络→网格→云计算

- **服务化趋势**：“打包式”软件→“服务式”软件
- **全球化生态化趋势**：一体化软件，协调共享软件
- **智能化：智能手机软件**

Good Software

- ✓ Meeting user's need
- ✓ Maintainability
- ✓ Software must evolve to meet changing needs
- ✓ Dependability
  - Software must be trustworthy
- ✓ Efficiency
  - Software should not make wasteful use of system resources
- ✓ Usability

1.4 Software Crisis

随着软件规模不断扩大软件危机显现

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。

这类问题绝不仅仅是“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类问题。

概括来说，软件危机包含两方面问题：

其一：如何开发软件，以满足不断增长、日趋复杂的需求；

其二：如何维护数量不断膨胀的软件产品。

软件事故

- 1963年，美国用于控制火星探测器的计算机软件中的一个“，”号被误写为“.”，而致使飞往火星的探测器发生爆炸，造成高达数亿美元的损失。
- 2002年阿丽亚娜火箭爆炸
- 千年虫问题
- Windows offices
- 工业控制
- ……

举例1：ARIANE 5 火箭

1996年6月4日，Ariane 5火箭在发射37秒之后偏离其飞行路径并突然发生爆炸， 当时火箭上载有价值5亿美元的通信卫星。

原因：

- 程序中试图将64位浮点数转换成16位整数时产生溢出
- 缺少对数据溢出的错误处理程序
- 备份软件通过复制而成

举例2：Windows Vista系统

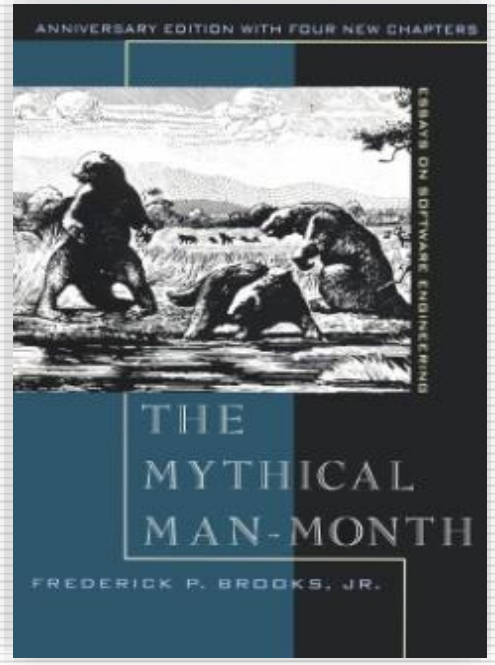
- 该系统从2001年开始研发，整个过程历时5年半，先后有9000位开发人员投入其中，耗资60亿美元，代码规模超过5000万行。
- 按照微软公司最初的计划，该系统面世时间应该在2003年，之后推迟到2004年下半年再到2005年初，最终在取消一些高级功能后于2006年11月正式发布。
- 由于整个系统过于庞杂，其开发管理相当混乱，以致于很多时间用在互相沟通和重新决定上。
- 从Vista Beta 1进入公开测试以来，程序错误总数已经超过2万个，这其中还不包括微软内部未公开的一些错误。

举例3：12306 网络购票系统

- 12306网络购票系统历时两年研发成功，耗资3亿元人民币，于2011年6月12日投入运行。
- 2012年1月8日春运启动，9日网站点击量超过14亿次，出现网站崩溃、登录缓慢、无法支付、扣钱不出票等严重问题。
- 2012年9月20日，由于正处中秋和“十一”黄金周，网站日点击量达到14.9亿次，发售客票超过当年春运最高值，出现网络拥堵、重复排队等现象。

一个值得思考的问题

为什么一个看似简单的东西，
却很有可能变成一个落后进度
、超出预算、存在大量缺陷的
怪物?



Brooks, F. P., *The Mythical Man Month*, Addison-Wesley, 1975

《人月神话》源于Brooks在IBM公司担任OS/360系统项目经理时的实践经验

Phenomena of Crisis

- ✓ **Late**

  schedule and plan cannot be controlled

- ✓ **Large cost**

  over budget

- ✓ **Poor quality**

  faulty, bug

- ✓ **Difficult to maintain**

  can’t see it and restriction by hardware

- ✓ **Low productivity**

Why?

- ✓**invisible**
- ✓**huge size**
- ✓ **too complicated:** logic, not governed by physical law, lack of natural constrains
- ✓**so flexible**
- ✓**more factor** to affect software quality: person ability, tean relation, product complexity, available time, other
- ✓**unclear requirement**
- ✓**error recognition**

软件危机的原因

产生软件危机的原因可以归纳为主、客观两个方面:

从客观上来看，软件不同于硬件，它的生产过程和产品都具有明显的“不可视”特征。（1）对于开发软件的过程进行管理和控制比较困难。在软件工程的早期，制定详细的开发计划并且进行全程跟踪调控，可望在一定程度上克服“开发过程不可视”造成的消极影响。（2）软件运行过程中如果发现了错误，那么必然是遇到了在开发时期(分析、设计、编码过程)引入的。利用足够的文档资料使不可视的产品可视化，有助于提升软件产品的可理解性和可维护性。

从主观上分析，导致软件危机发生的另一大原因，可以归于在计算机系统发展的早期，软件开发的“个体化”特点，主要表现为忽视软件需求分析的重要性、忽视软件的可理解性、文档不完备、轻视软件的可维护性、过分强调编码技巧等等方面。

1.5 What's Software Engineering

1968年，在德国，NATO的计算机会议上分析了软件危机的原因，探索用工程的方法进行软件开发和维护的可能性，提出了软件工程概念。

“软件工程”从此诞生

An original definition

Fritz Bauer, seminal conference of SE, 1968

"[Software engineering] is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines".

"软件工程是为了经济地获得可靠的和能在实际机器上高效运行的软件而确立和使用的健全的工程原理（方法）"。

IEEE Definition

Software engineering is an engineering discipline which is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.

软件工程是开发、运行、维护和修复软件的系统方法”。

软件工程的定义（书）

软件工程是采用工程的观念、原理、技术和方法来开发与维护软件，把经过时间考验而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。

软件工程的定义（学科）

软件工程是应用计算机科学、管理科学等原理开发软件的学科。它借鉴传统工程的原则和方法，以提高软件质量，降低开发和维护成本为目的学科（专业、技术）。

什么是软件工程（目的）

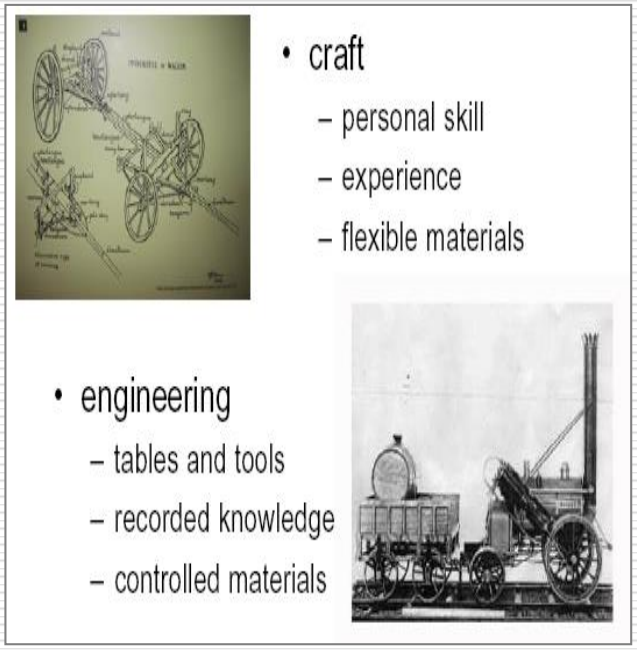
软件工程是 ① 将系统性的、规范化的、可定量的方法应用于软件的开发、运行和维护，即工程化应用到软件上；② 对①中所述方法的研究。

软件工程的基本目标：

- **较低的开发成本**
- **按时完成开发任务并及时交付**
- **实现客户要求的功能**
- **所开发软件具有良好的性能**
- **较高的可靠性、可扩展性、可移植性**
- **软件维护费用低**

工程的方法

工程是将理论和所学的知识应用于实践的科学，以便经济有效地解决实际问题。



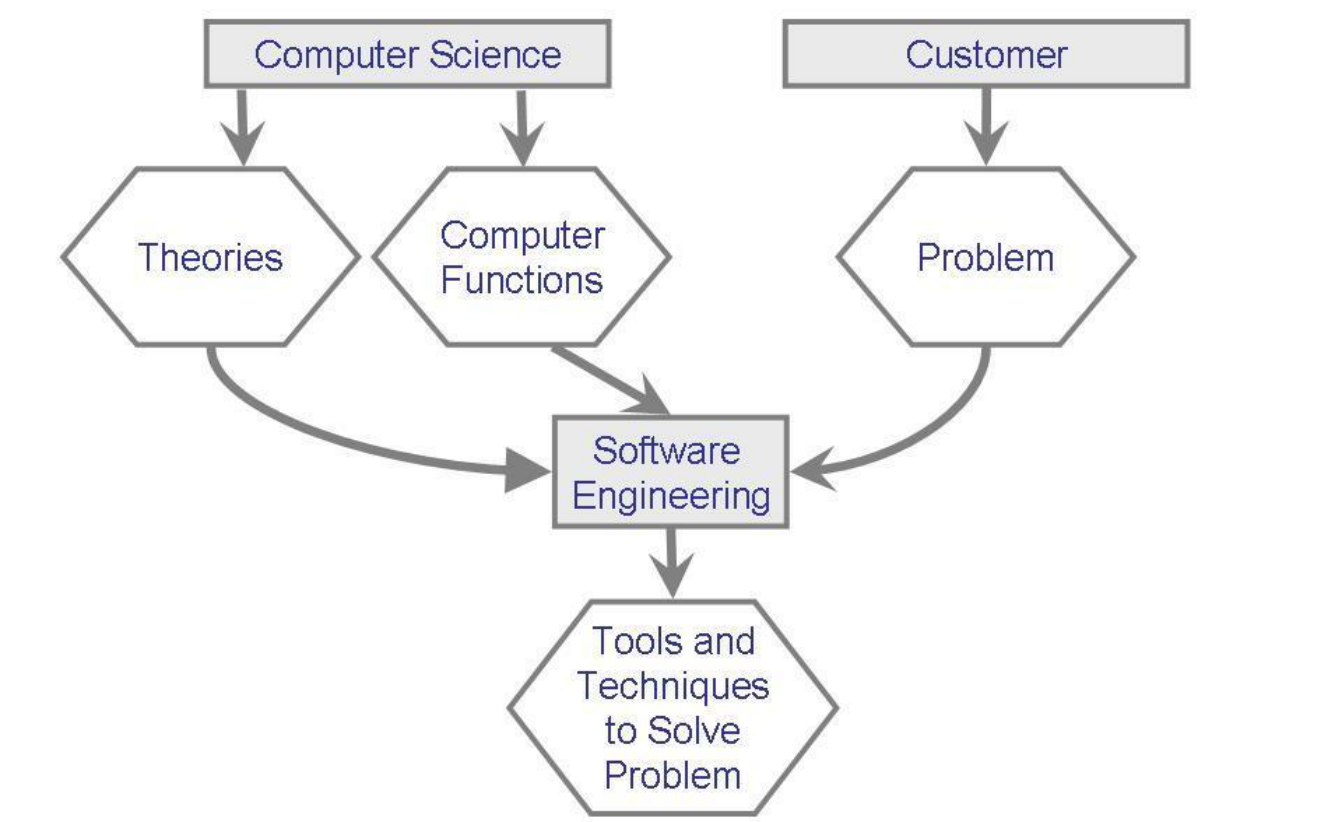
- **规模上的差异**
  - **花园小道 vs. 汽车高速公路**
  - **树上小屋 vs. 摩天大楼**
  - **加法程序 vs. 医院档案系统**
- **手工（Craft）：小规模的设计与建造**
  - **简单问题与单一目标**
  - **个人控制与个人技能**
- **工程（Engineering）：大规模的设计与建造**
  - **复杂问题与目标分解**
  - **团队协作，需要考虑运营、管理、成本、质量控制、安全等**

软件工程与计算机科学的区别

科学是发现世界上已经存在的事物，回答“是什么”和“为什么”的问题。
工程是创造世界上从未存在的事物，回答“做什么”和“怎么做”的问题。

科学	工程
科学是我们认识自然的过程	工程是我们征服自然的过程
科学解决的是“为什么”的问题	工程解决的是“怎么办”的问题
科学需要独立思考与自由探索	工程需要纪律性与团队精神
科学具有不确定性	工程具有计划性与实用性

软件工程与计算机科学的区别



计算机科学研究构成计算机本身的理论和结构，诸如硬件设计或者算法的理论证明等。

软件工程将计算机作为问题求解的**工具**，**设计**和**实施**尚未存在的方案用以服务社会。

在计算机学科位置

计算机科学
技术

计算学科
(Computing discipline)

- 计算机科学 (CS)
- 计算机工程 (CE)
- 软件工程 (SE)
- 信息系统 (IS)

SE：A Cross-Discipline

- **a broad range of skills**
  - ✓ **Mathematics**
  - ✓ **Computer Science**
  - ✓ **Economics**
  - ✓ **Management**
  - ✓ **Psychology**
- **applied to all phases**

软件工程五要素

项目

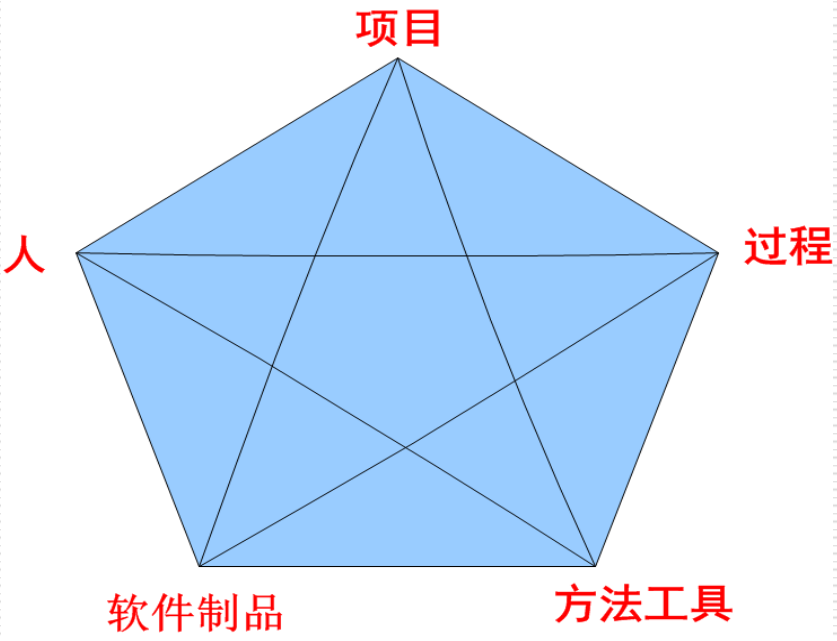
根据开发系统的特点和要求选用不同特长的人组成项目团队，采用适宜的组织方式，开发方法、工具和过程开发出用户需要的软件制品。

管理者

对项目进行管理和控制，人员组织、计划制定、成本估算、跟踪与控制、质量保证、配置管理等。

过程

过程模型、过程活动、规范和标准、软件度量、项目管理、配置和变更管理、过程改进等。



Two main research aspects

- 软件开发技术：　软件开发方法学
  软件开发过程
  软件工具和软件工程环境
- 软件工程管理：　软件管理学
  软件经济学
  软件心理学

软件开发环境

软件开发环境是相关的一组软件工具的集合，它支持一定的软件开发方法或按一定的软件开发模型组织而成。（IDE）

程序设计环境，系统合成环境，项目管理环境。

软件工具

- 软件工具是指支持计算机软件开发、维护、模拟、移植和管理而研制的程序系统。
- 机械工具能放大人的体能，
- 软件工具能放大人的智能。
- 类型：开发，模拟，测试，评估，运行，维护，性能测量，程序设计支持等。

CASE-计算机辅助软件工程

- CASE 是一组工具和方法的集合，可以辅助软件开发生命周期各阶段进行软件开发。
- 支持设计，开发，管理
- IBM Rational Rose
- 数据库设计工具

软件工程不是编程

Small project	Huge project
You	Teams
Build what you want	Build what they want
One product	Family of products
Few sequential changes	Many parallel changes
Short-lived	Long-lived
Cheap	Costly
Small consequences	Large consequences

Programming Engineering

Programming → Engineering

- **People**
  - Who else would do the work?
  - Range from novice to very experienced
- **Processes**
  - To organize and manage the efforts of individuals
  - Range from informal to very formal
- **Tools**
  - To support the people and the processes
  - Range from simple to very advanced

People + Processes + Tools ⇒ Product

“软件工程”课程教学的小目标

- ✓ 转变对软件开发的认识：

程序 —上升→ 系统

- ✓ 转变思维定式：

程序员 —上升→ 系统工程师（系统分析员）

- ✓ 进行工程化训练，学习写文档，学习标准化开发

1.6 Software life cycle

软件产品或软件系统从设计、投入使用、到被淘汰的全过程。

Phases of software life cycle

(1) problem def.

(2) feasibility study

(3) requirement analysis

(4) architecture design

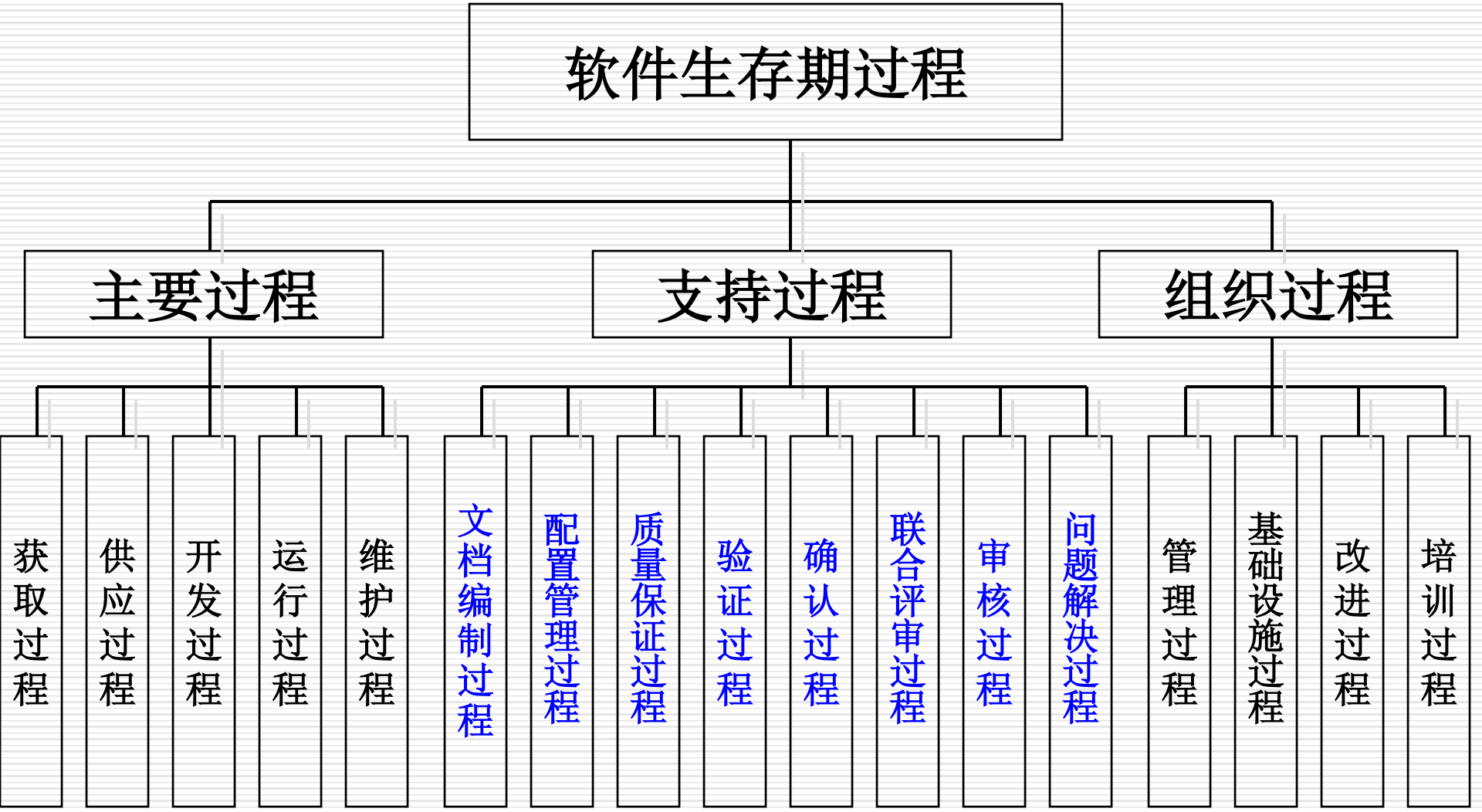
(5) detailed design

(6) coding and implementation

(7) testing

(8) run and maintenance

新的国际标准定义的软件生存过程（1995 ISO/IEC 12207）



1.7 Software Process Model

Software process is a set of activities and associated results which produce a software product.

Software process model is a simplified description of a software process which is presented from a particular perspective.

Abstraction, framework

Software Process Model

- waterfall model
- incremental model
- iterative model
- rapid prototyping model
- automatic transformation model
- ……