

软件工程专业导论

课程介绍

学习目的：

- ✓ 使学生初步了解软件工程专业学科
- ✓ 认识软件工程专业的知识内容与学习方法
- ✓ 为进入软件工程专业学习打下良好基础

课时安排：

- ✓ 课堂讲授 32学时

考核方法：

- ✓ 期末笔试 (80%)： 中国大学MOOC - 软件工程专业导论占30%
- ✓ 平时成绩 (20%)

课程介绍

参考教材：

- ✓ 孙家广等，《软件工程——理论、方法与实践》，高等教育出版社，2008年
- ✓ 战德臣等，《大学计算机-计算思维导论》，电子工业出版社，2013年
- ✓ 朱少民等，《软件工程导论》，清华大学出版社，2009年
- ✓ Shari Lawrence Pfleeger, Software Engineering-- Theory and Practice (4th Edition), Prentice Hall, 2009
- ✓ Roger S. Pressman, Software Engineering—A Practitioner's Approach (7th Edition), McGraw-Hill Education, 2007 (中文翻译版，机械工业出版社，2011)
- ✓ 其他“软件工程导论”或“软件工程概论”的参考书



初识软件工程专业

学习目标：

- 了解软件工程的一些基本概念
- 了解软件工程的发展
- 了解软件工程专业知识体系和课程体系



1. 什么是软件（Software）？

- 软件是一系列按照**特定顺序组织的计算机数据和指令**，是计算机中的非有形部分。
- 软件并不一定只包括可以在计算机上运行的计算机程序，有些定义中，与计算机程序相关的文档，一般也被认为是软件的一部分。简单的说软件就是**程序加文档的集合体**。

-----维基百科（Wikipedia）

1. 什么是软件（Software）？

如操作系统、编译器与连接器、网络管理系统、数据库管理系统等

系统软件（System Software）：最靠近计算机硬件的一层软件 —— 控制和协调计算机及外部设备、支持应用软件开发与运行的软件。

1. 什么是软件（Software）？

如网络构件、中间件及软件运行平台、软件开发环境及工具、各种接口软件与工具组等

支撑软件（Support Software）： 软件系统的中间层，支撑各种软件的开发、运行与维护的软件。

系统软件（System Software）： 最靠近计算机硬件的一层软件 —— 控制和协调计算机及外部设备、支持应用软件开发与运行的软件。

1. 什么是软件（Software）？

应用软件（Application Software）：为满足特定应用领域、不同应用问题之需求的专用软件。

支撑软件（Support Software）：支撑各种软件的开发、运行。

如行业应用软件、文字与媒体处理软件、办公自动化软件、企业管理软件、工业控制软件、各类嵌入式应用软件等

系统软件（System Software）：最靠近计算机硬件的一层软件 —— 控制和协调计算机及外部设备、支持应用软件开发与运行的软件。

2. 什么是工程（Engineering）？

- 工程是科学和数学的某种应用，通过这一应用，使自然界的物质和能源的特性能够通过各种结构、机器、产品、系统和过程，以最短的时间和最少的人力、物力做出高效、可靠且对人类有用的东西。将自然科学的理论应用到具体工农业生产部门中形成的各学科的总称。

-----百度百科

- The term engineering is derived from the Latin ingenium, meaning "cleverness" and ingeniare, meaning "to contrive, devise".

-----Wikipedia

2. 什么是工程（Engineering）？



工程领域列表

- 工程科学
- 系统工程
- 工业工程
- 材料工程
- 建筑工程
- 土木工程
- 机械工程
- 航空工程
- 航海工程
- 电机工程
- 软件工程
- 化学工程
- 生物工程

3. 什么是软件工程（Software Engineering）？



软件工程

- **软件工程（经典定义）：**应用计算机科学理论和技术以及工程管理原则和方法，按预算和进度实现满足用户要求的软件产品的工程，或以此为研究对象的学科。
- **软件工程主要研究内容：**软件开发过程、软件开发方法、软件工程管理与支持、软件质量保障、软件工程度量、计算机辅助软件工程环境及工具等。

（引自《计算机科学技术百科全书》，清华大学出版社，2005.11）

- **IEEE CS定义：**软件工程是 ①将系统化的、学科的、可量化的方法应用于软件的开发、运行和维护，即针对软件的应用；②对于上述应用方法的研究。

3. 什么是软件工程（Software Engineering）？

软件工程

➤ **软件工程（经典定义）：**应用计算机科学理论和技术以及工程管理原则和方法，按预算和进度实现满足用户要求的软件，**过程：**生产一个最终能满足需求且达到工程目标的软件产品所需要的步骤。

目标：生产具有正确性、可用性以及成本合适的软件产品。

原则：围绕工程设计、工程支持以及工程管理在软件开发过程中必须遵循的原则。

- **软件工程的框架：**目标、过程和原则
- **软件工程的要素：**方法、工具和过程

3. 什么是软件工程（Software Engineering）？

软件工程

➤ **软件工程（经典定义）：**应用计算机科学理论和技术以及工程管理原则和方法，按预算和进度实现满足用户要求的软件产品的工程，或以此为研究对象的学科。

软件工程主要研究内容

方法：软件开发“如何做”的工程方法、原理与技术。

工具：软件工程方法的自动化或半自动化的支撑。

过程：软件工程方法与工具在软件开发与运行各阶段的运用。

➤ **软件工程的框架：**目标、过程和原则

➤ **软件工程的要素：**方法、工具和过程

4. 软件工程的发展和趋势

- 世界上第一个程序员
- 现代计算机软件的出现
- 软件危机
- 软件工程的提出
- 传统软件工程
- 现代软件工程

4. 软件工程的发展和趋势

- 世界上第一个程序员



- 1842-1843年，Ada翻译了意大利数学家luigi Menabrea的文章，并在注记中用Babbage分析机的指令写下如何计算Bernoulli数的详细步骤。历史学家们认为这注记是历史上第一个程序
- 1980年12月10日，美国国防部以她的名字命名了他们的语言并颁布了Ada语言标准

4. 软件工程的发展和趋势

- 现代计算机软件的出现

- 20世纪50年代，软件伴随着第一台电子计算机的问世诞生了。以写软件为职业的人也开始出现，他们多是经过训练的数学家和电子工程师。
- Colossus计算机（1943年，Tommy Flowers领导发明）
- ENIAC计算机（1946年，John Mauchly领导发明）
- EDVAC计算机（1948年，Von Neumann领导发明）

4. 软件工程的发展和趋势

- 现代计算机软件的出现
 - 硬件更新速度快，与之配套的软件也需要更新
 - 软件由使用该软件的个人或机构研制，带有强烈的个人色彩。软件开发没有系统的方法可以遵循，除了源代码往往没有软件说明书等文档

4. 软件工程的发展和趋势

● 软件危机

- 从60年代中期到70年代中期，软件开始作为一种产品被广泛使用，出现了“软件作坊”专职应别人的需求写软件
- 但软件的数量急剧膨胀，软件需求日趋复杂，维护的难度越来越大，开发成本令人吃惊地高，而失败的软件开发项目却屡见不鲜。“软件危机”就这样开始了！

4. 软件工程的发展和趋势

- 软件危机

- 预算超出、财产损失、死亡事故

- OS/360操作系统是由上世纪六十年代开始的一个庞大复杂的程序，投入到这个项目中的软件工程师超过2000人（Windows2000也只动用了1700名），后来，Fred Brook在《The Mythical Man-Month》中承认他们开发前犯了一个错误，没有开发一个有逻辑概念的体系设计，从而浪费了好几百万美元

4. 软件工程的发展和趋势

- 软件工程的提出

- 1968年10月，北大西洋公约组织NATO召开计算机科学会议，Fritz Bauer首次提出“软件工程”概念及克服“软件危机”的策略，强调按照工程化原则和方法组织软件开发工作软件工程技术领域由此应运而生

4. 软件工程的发展和趋势

1946-1956: 程序设计时代; 个体手工制作, 采用机器语言/汇编语言编程, 主要依靠个人编程技巧。

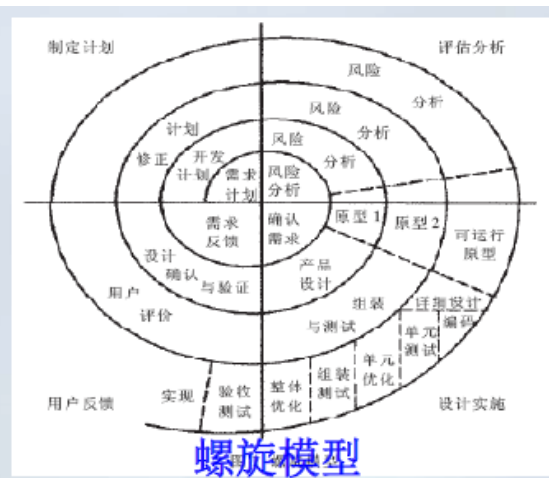
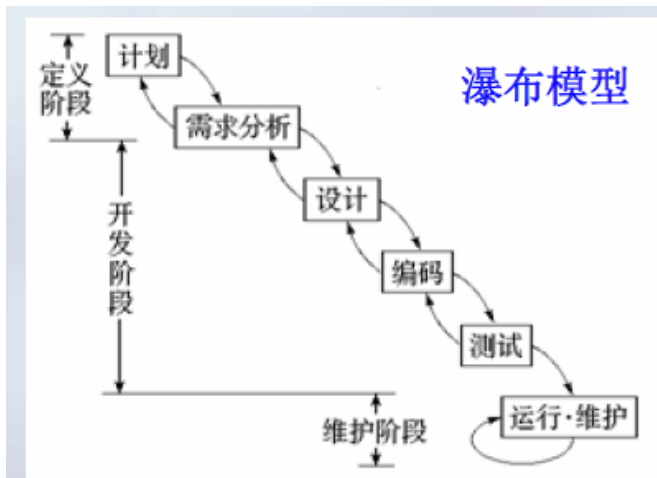
1956-1968: 程序系统时代; 作坊式小团队合作, 采用高级语言编程, 以个人编程技巧为主, 开始有结构化方法。

1968以来, 软件工程时代; 工程化生产方式, 采用软件工程方法, 并逐步发展形成了结构化方法、面向数据结构方法、面向对象的方法、构件化方法、面向服务的方法等。

4. 软件工程的发展和趋势

● 传统软件工程

- 一是从管理的角度，希望实现软件开发过程的工程化
 - “瀑布式”生命周期模型
 - 螺旋模型



4. 软件工程的发展和趋势

- 传统软件工程

- 二是侧重与对软件开发过程中分析、设计的方法的研究

- 结构化程序设计方法

```
#include <stdio.h>
#define N 4
main() {
    //printf("Hello!" + CITY " NC");
    int D[N][N];
    int S[N];
    int Sum;
    int i;
    int j;
    int K;
    int L;
    int Qtemp;
    int Found;

    S[0]=0;
    Sum=0;
    Q[0][1]=2;
    Q[0][2]=6;
    Q[0][3]=5;
    Q[1][0]=2;
    Q[1][2]=4;
    Q[1][3]=4;
    Q[2][0]=6;
    Q[2][1]=4;
    Q[2][3]=2;
    Q[3][0]=5;
    Q[3][1]=4;
    Q[3][2]=2;

    i=1;
    do {
        K=1;
        Qtemp=10000;
        do {
            L=0;
            Found=0;
            do {
                if(S[L]==K) {
                    Found=1;
                    break;
                }
                else L++;
            }
            while(L<K);
            if(Found==0 && D[K][L]>Qtemp) {
                j=K;
                Qtemp=D[K][L];
            }
            K++;
        }
        while(K<N);
        S[i]=j;
        Sum+=Q[i][j];
        i++;
    }
    printf("total length:%d",Sum);
}
```

主函数0

数据结构与数据

算法(的实现)

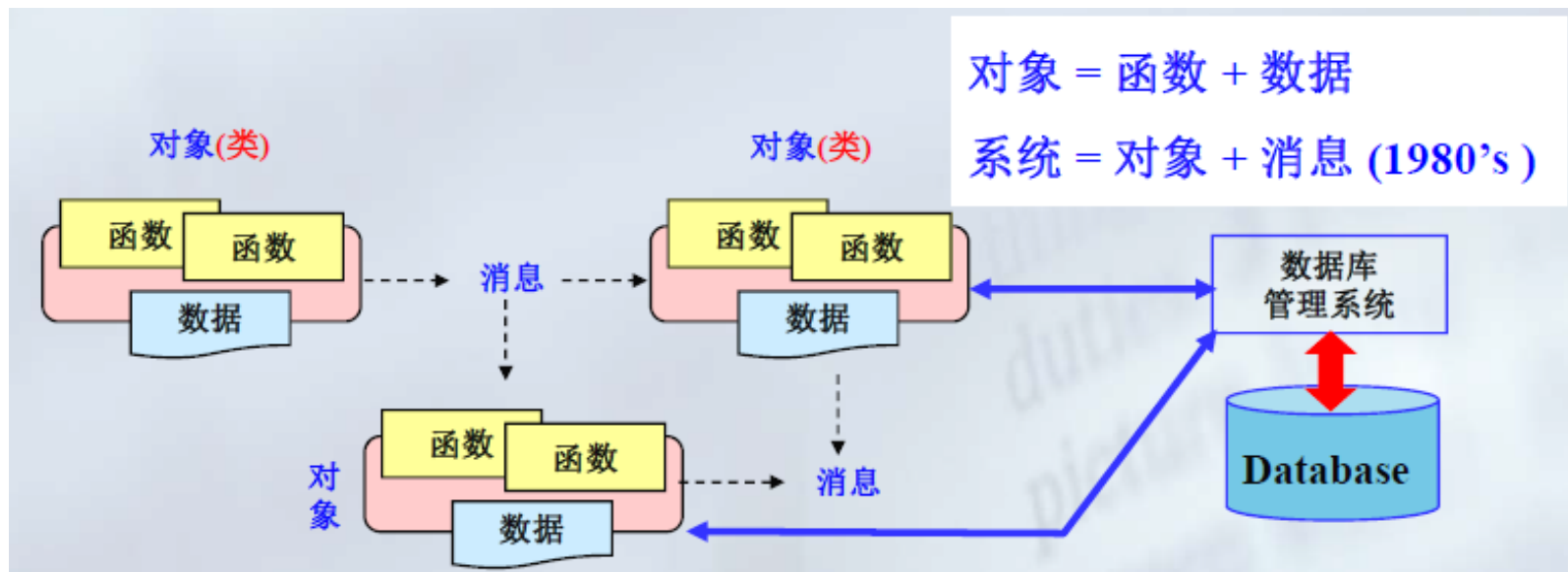
函数0

函数0

4. 软件工程的发展和趋势

- 现代软件工程

- 强调人和机器在不同层次的不循环发展的关系
- 面向对象的分析、设计方法（OOA和OOD）



4. 软件工程的发展和趋势

- 现代软件工程

- 强调人和机器在不同层次的不循环发展的关系

- 与之相应的是从企业管理的角度提出的软件过程管理

- 美国卡内基梅隆大学软件工程研究所（SEI）建立的CMM（Capability Maturity Model），即能力成熟度模型

4. 软件工程的发展和趋势

● 软件工程方法发展历程

- 1960's-1970's: 结构化方法

- 1980's: 面向对象的方法

- 1990's: 构件化方法和Web Services

- 2000's: 面向服务的SOA方法

- 2010's: 基于互联网与云计算的软件开发方法

4. 软件工程的发展和趋势

阶段1: 独立的程序(Independent Programming Service)

- **1950s-1960s:** 机器为中心, 专业服务公司, 如IBM

阶段2: 软件产品(Software Product)

- **1960s-1970s:** 软件业独立发展, 软件产品公司, 如ADR

阶段3: 企业解决方案(Enterprise Solution)

- **1970s-1980s:** 应用为中心, 企业解决方案公司, SAP, Oracle

阶段4: 支持大众应用的软件包(Packaged Software for Mass)

1980s-1994: 个人为中心, 大众软件公司, Microsoft, Apple

阶段5: 网络软件与服务(Internet - Based Software and Service)

- **1990s,** 网络为中心, 互联网公司, Netscape, Yahoo, Google
- **2000s,** 社会为中心, 服务公司, Facebook, Twitter, YouTube

5. 软件工程技术体系

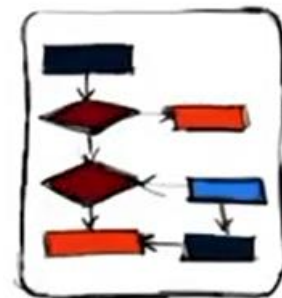
- 软件工程的研究对象

- 软件 + 工程：

- 理论、技术、管理、方法；目标、原则、活动

- 软件工程的关注点：

- 软件开发过程
 - 软件开发方法
 - 软件开发工具
 - 软件开发质量



5. 软件工程技术体系

● 软件工程科学技术范畴

- 软件过程（Software Process）
- 软件开发方法（Software Development Method）
- 软件需求工程（Software Requirement Engineering）
- 软件体系结构（Software Architecture）
- 软件开发工具与环境（Software Development Environment）
- 软件复用与软构件（Software Reuse & Component）
- 软件中间件（Software Middleware）
- 软件测试（Software Testing）
- 软件维护（Software Maintenance）

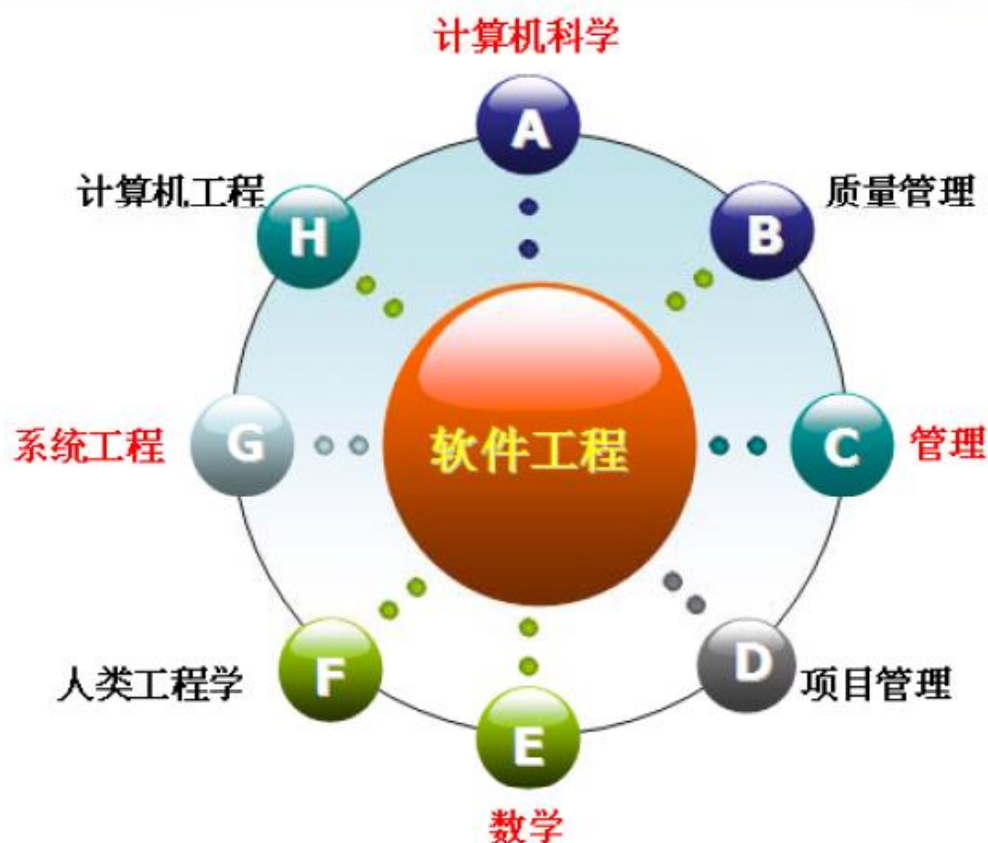
5. 软件工程技术体系

- 软件工程科学技术范畴

- 软件过程度量（Software Process Measurement）
- 软件工程管理（Software Engineering Management）
- 软件质量(Software Quality)
- 软件可信性（Software Dependability）
- 软件服务工程（Software Service Engineering）
- 领域工程（Domain Engineering）
- 软件逆向工程（ Software Reverse Engineering ）
-

5. 软件工程技术体系

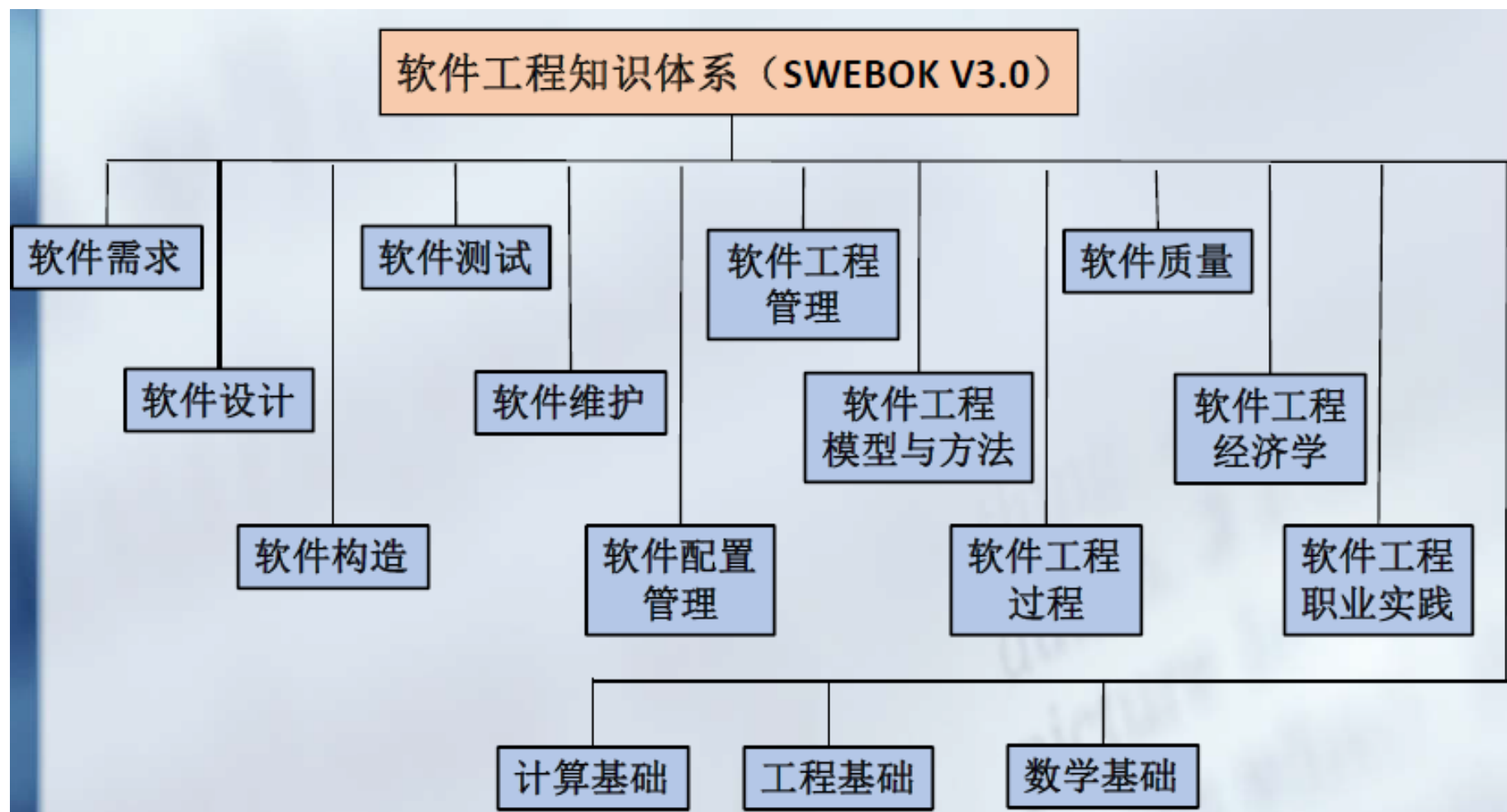
- 软件工程相关学科



6.软件工程专业知识体系

- IEEE CS 软件工程专业知识体系SWEBOK
 - 美国电子电气工程师学会IEEE CS与美国计算机联合会ACM成立了软件工程协调委员会，于1994年开始研究软件工程知识体系（SWEBOK）
 - 2001、2004年先后发布SWEBOK 1.0， 2.0版
 - 2014年3月，正式发布SWEBOK 3.0版，成为软件工程知识体系的样板

6.软件工程专业知识体系



7.软件工程专业课程体系		
第一学年		
秋季学期	春季学期	小学期
几何代数	离散数学	
工科数学	工科数学	
	数字电路	
专业导论	集合图论	
C语言	C++编程 编程实践	

7.软件工程专业课程体系

第一学年			第二学年		
秋季学期	春季学期	小学期	秋季学期	春季学期	小学期
几何代数	离散数学				
工科数学	工科数学			概率统计	
	数字电路		组成原理	接口技术	
专业导论	集合图论		数据结构	操作系统	
			结构课设	数据库	
			图形学	数库课设	
C语言	C++编程		Java语言	UML语言	
	编程实践				
				软件工程	工业实训
				工程课设	

公共基础课：大学英语、思想品德、人文社科、毛邓理论、体育、其它公共学修课。

7.软件工程专业课程体系

第一学年			第二学年			第三学年		
秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋季学期	春季学期	小学期
几何代数	离散数学			概率统计				
工科数学	工科数学			接口技术				
	数字电路		组成原理	操作系统				
专业导论	集合图论		数据结构	数据库		网络	编译原理	
			结构课设	数据库			算法导论	
			图形学	数据库课设				
C语言	C++编程		Java语言	UML语言		.NET	J2EE	
	编程实践					体系结构	方向选修	
						系统分析		
				软件工程	工业实训	项目管理	软件质量	综合课设
				工程课设				

公共基础课：大学英语、思想品德、人文社科、毛邓理论、体育、其它公共学修课。

7.软件工程专业课程体系

第一学年			第二学年			第三学年			第四学年
秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋春学期
几何代数 工科数学	离散数学 工科数学				概率统计	数学课			
	数字电路		组成原理	接口技术					
专业导论	集合图论		数据结构 结构课设	操作系统 数据库 数据库课设		网络	编译原理 算法导论		毕业设计
C语言	C++编程 编程实践		Java语言	UML语言		.NET 体系结构 系统分析	J2EE 方向选修		
				软件工程 工程课设	工业实训	项目管理	软件质量	综合课设	工业实习

公共基础课：大学英语、思想品德、人文社科、毛邓理论、体育、其它公共学修课。

7.软件工程专业课程体系

第一学年			第二学年			第三学年			第四学年
秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋春学期
几何代数	离散数学								
工科数学	工科数学			概率统计					
	数字电路		组成原理	接口技术					
专业导论	集合图论		数据结构	操作系统		网络	编译原理		
			结构课设	数据库			算法导论		
			图形学	数据库课设					毕业设计
C语言	C++编程		Java语言	UML语言		.NET	J2EE		
	编程实践					体系结构	方向选修		
						系统分析			
				软件工程	工业实训	项目管理	软件质量	综合课设	工业实习
				工程课设					

硬件课

7.软件工程专业课程体系

第一学年			第二学年			第三学年			第四学年
秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋春学期
几何代数	离散数学			概率统计		专业基础课：理论课+系统课			
工科数学	工科数学			接口技术					
	数字电路		组成原理						
专业导论	集合图论		数据结构 结构课设	操作系统 数据库 数据库课设		网络	编译原理 算法导论		毕业设计
C语言	C++编程 编程实践		Java语言	UML语言		.NET 体系结构 系统分析	J2EE 方向选修		
				软件工程 工程课设	工业实训	项目管理	软件质量	综合课设	工业实习

公共基础课：大学英语、思想品德、人文社科、毛邓理论、体育、其它公共学修课。

7.软件工程专业课程体系

第一学年			第二学年			第三学年			第四学年
秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋春学期
几何代数	离散数学			概率统计					
工科数学	工科数学			接口技术					
	数字电路		组成原理	操作系统					
专业导论	集合图论		数据结构	数据库		网络	编译原理		
			结构课设	数据库			算法导论		
			图形学	数库课设					毕业设计
C语言	C++编程		Java语言	UML语言		.NET	J2EE		
	编程实践					体系结构	方向选修		
						系统分析			
专业技术课				软件工程	工业实训	项目管理	软件质量	综合课设	工业实习
				工程课设					

公共基础课：大学英语、思想品德、人文社科、毛邓理论、体育、其它公共学修课。

7.软件工程专业课程体系

第一学年			第二学年			第三学年			第四学年
秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋春学期
几何代数	离散数学			概率统计					
工科数学	工科数学			接口技术					
	数字电路		组成原理	操作系统					
专业导论	集合图论		数据结构	数据库		网络	编译原理		
			结构课设	数据库			算法导论		
			图形学	数据库课设					毕业设计
C语言	C++编程		Java语言	UML语言		.NET	J2EE		
	编程实践					体系结构	方向选修		
						系统分析			
软件工程管理课				软件工程	工业实训	项目管理	软件质量	综合课设	工业实习
				工程课设					

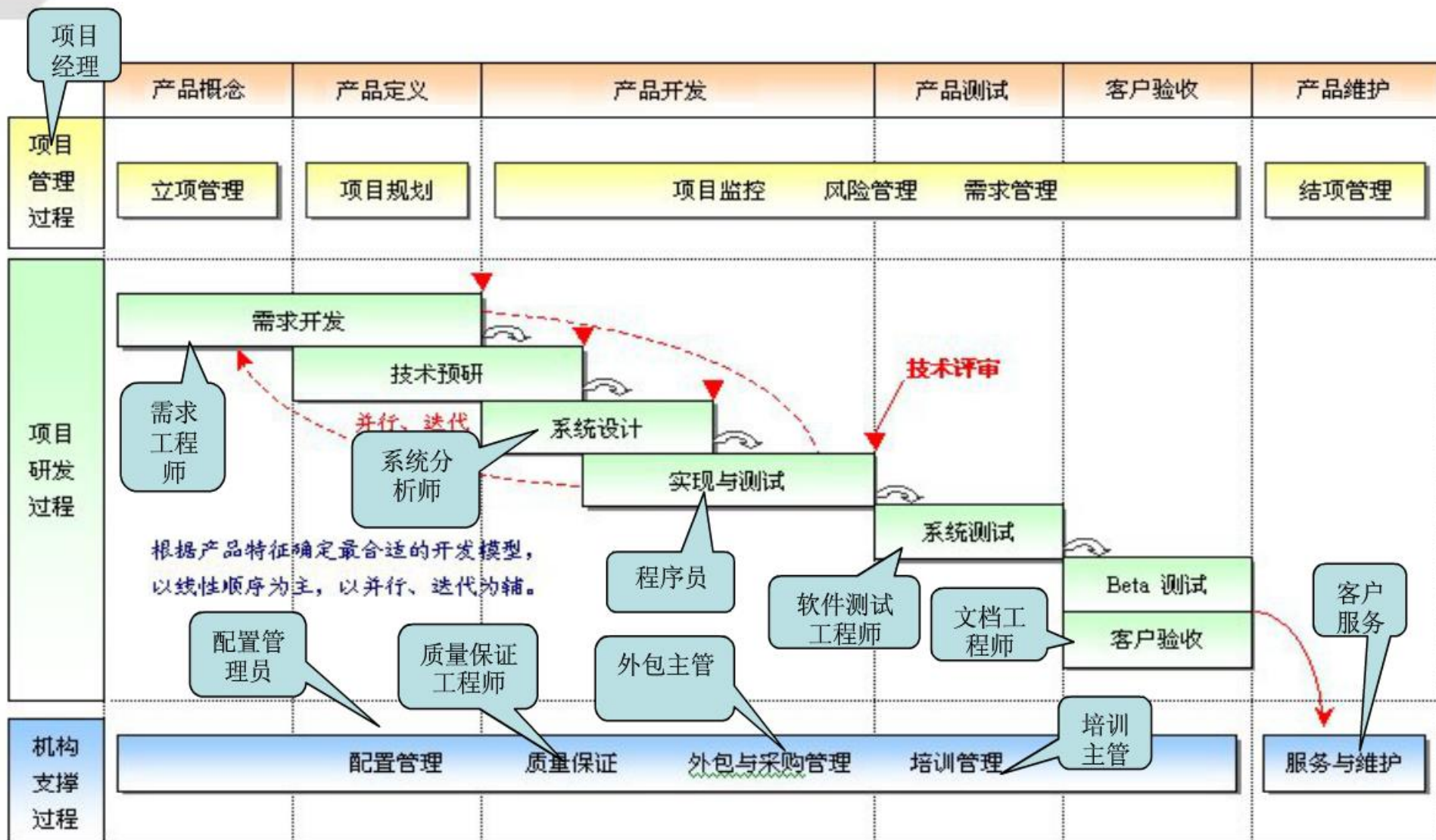
公共基础课：大学英语、思想品德、人文社科、毛邓理论、体育、其它公共学修课。

7.软件工程专业课程体系

第一学年			第二学年			第三学年			第四学年
秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋季学期	春季学期	小学期	秋春学期
几何代数 工科数学	离散数学 工科数学			概率统计		数学课			专业基础课
	数字电路		组成原理	接口技术		硬件课			
专业导论	集合图论		数据结构 结构课设	操作系统 数据库 数据库课设		网络	编译原理 算法导论		
			图形学						
C语言	C++编程 编程实践		Java语言	UML语言		.NET 体系结构 系统分析	J2EE 方向选修		毕业设计
专业技术课				软件工程 工程课设	工业实训	项目管理	软件质量	综合课设	工业实习
			软件工程管理课						

公共基础课：大学英语、思想品德、人文社科、毛邓理论、体育、其它公共学修课。

软件工程岗位概览



8. 如何学习软件工程专业知识

- 如何学习

- 过程是最好的老师
- 积极、认真、按时、高质量地完成大作业（编程实践），四年中至少独立完成5千-1万行编程。
- 积极参加俱乐部活动，至少完成1-2个项目。
- 积极参加工业实习

8. 如何学习软件工程专业知识

- 理论 -> 实践 -> 理论

百度

✓ 什么是...

✓ ...是如何工作的

✓ 如何理解...

Google

What is

How it works

How to understand

8. 如何学习软件工程专业知识

- 如何度过四年大学生活
 - 每天做好该做的事
 - 不要迷恋游戏
 - 培养专业的兴趣和爱好
 - 认真完成作业，尤其是编程实验课
 - 积极参加各项活动
 - 坚持锻炼身体