



SW01

概述

目标

- 什么是软件工程，为什么它很重要
- 软件工程入门的几个关键问题
- 软件生存周期(lift cycle)
- 道德和职业问题对于软件工程师的重要性

内容

- 软件
- 软件工程学
- 软件工程中的常见问题
- 软件生存周期
- 职业和道德上责任

一 软件

1.1 什么是软件

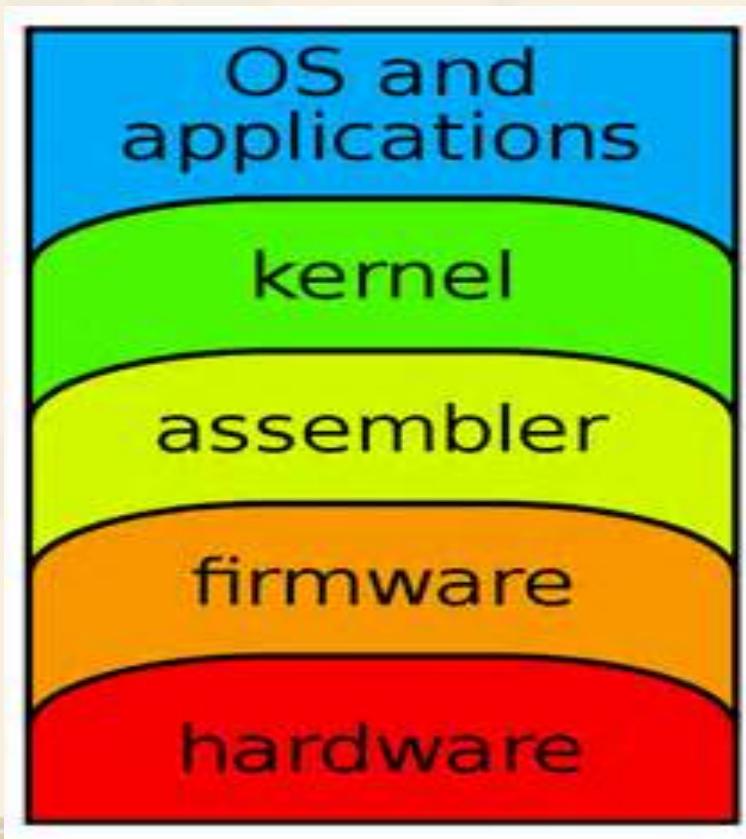
- 软件是由计算机程序、数据及文档组成。
- 程序是按事先设计的功能和性能要求执行的指令序列
- 数据是使程序能正常操纵信息的数据结构
- 文档是与程序开发，维护和使用有关的图文材料

1.2 软件特点

- 抽象性
- 无磨损性
- 对计算机硬件依赖性
- 软件的手工开发方式
- 软件本身复杂性
- 软件的高成本

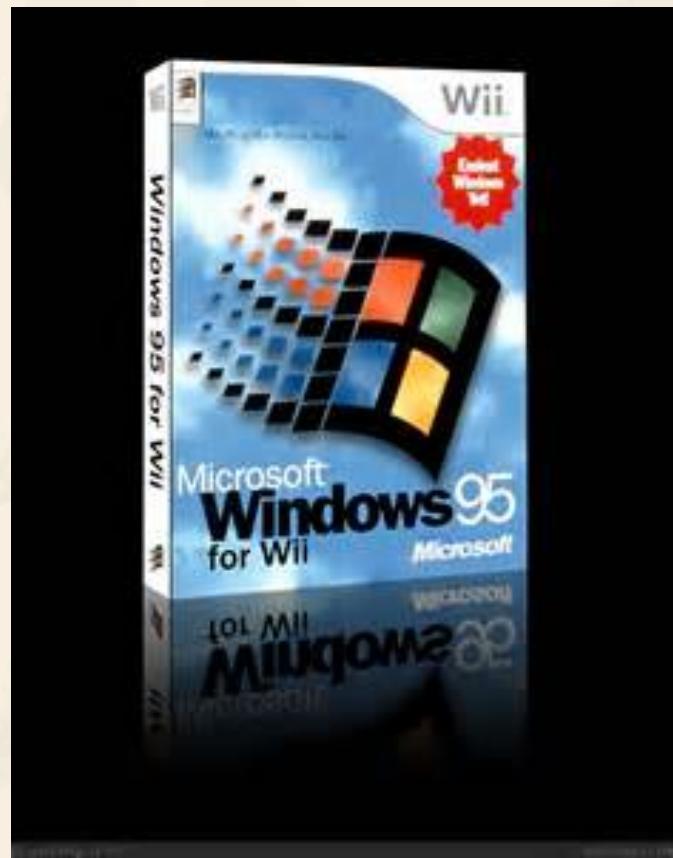
1. 2 软件特点

- 抽象性



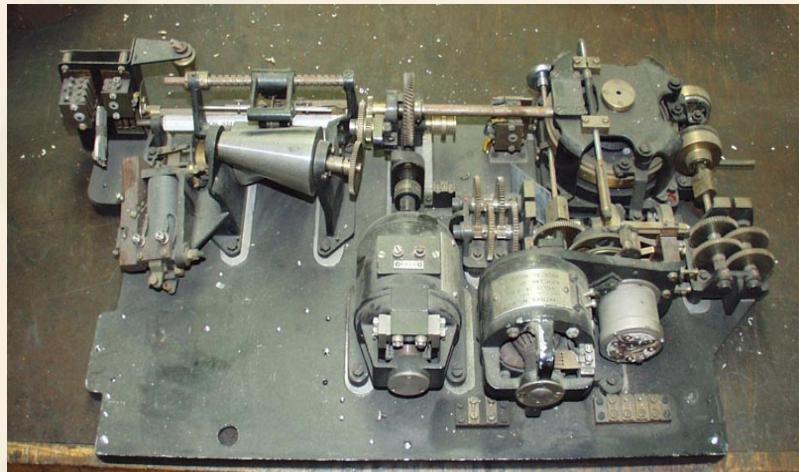
1.2 软件特点

- 抽象性
- 无磨损性
- 对计算机硬件依赖性
- 软件的手工开发方式
- 软件本身复杂性
- 软件的高成本



1.2 软件特点

- 抽象性
- 无磨损性
- 对计算机硬件依赖性
- 软件的手工开发方式
- 软件本身复杂性
- 软件的高成本



1.2 软件特点

- 抽象性
- 无磨损性
- 对计算机硬件依赖性
- 软件的手工开发方式
- 软件本身复杂性
- 软件的高成本



1.2 软件特点

- 抽象性
- 无磨损性
- 对计算机硬件依赖性
- 软件的手工开发方法
- 软件本身复杂性
- 软件的高成本

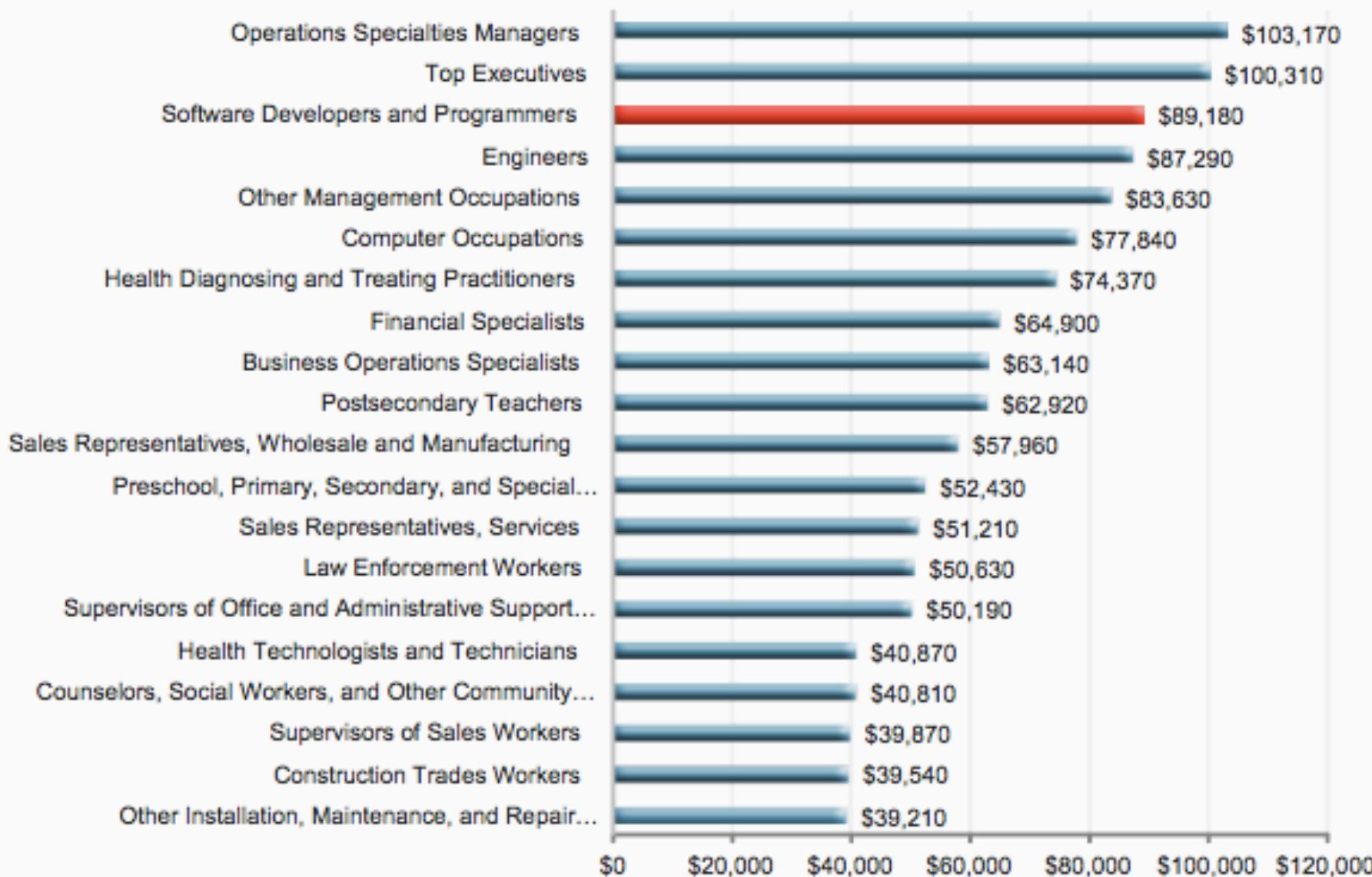


1.2 软件特点

- 抽象性
- 无磨损性
- 对计算机硬件依赖性大
- 软件的手工开发量小
- 软件本身复杂性高
- 软件的高成本



**Figure 8. Median Annual Income by Occupation, 2013:
Top 20 occupation groups with at least 1 million workers**



Source: Bureau of Labor Statistics, Occupational Employment Statistics .

1.3 软件分类

系统软件，支撑软件，应用软件

- 系统软件

操作系统、数据库管理系统(DBMS)、设备驱动程序、通信处理程序等

- 支撑软件

文本编辑程序、文件格式化程序、磁盘向磁带数据传输的程序、支持需求分析、设计、实现、测试和支持管理的软件

- 应用软件

商业数据处理软件、工程与科学计算软件、计算机辅助设计 / 制造软件、系统仿真软件、智能产品嵌入软件、医疗、制药软件、办公自动化软件

1.4 软件的发展

- 软件的发展经历了三个阶段
 - 程序设计阶段 — 50至60年代



1.4 软件的发展

■ 软件的发展经历了三个阶段

▶ 程序系统阶段 — 60至70年代

这一时期主要围绕软件项目，开展了开发模型、支持工具以及开发方法的研究。如：瀑布模型、结构化方法（自顶向下）等

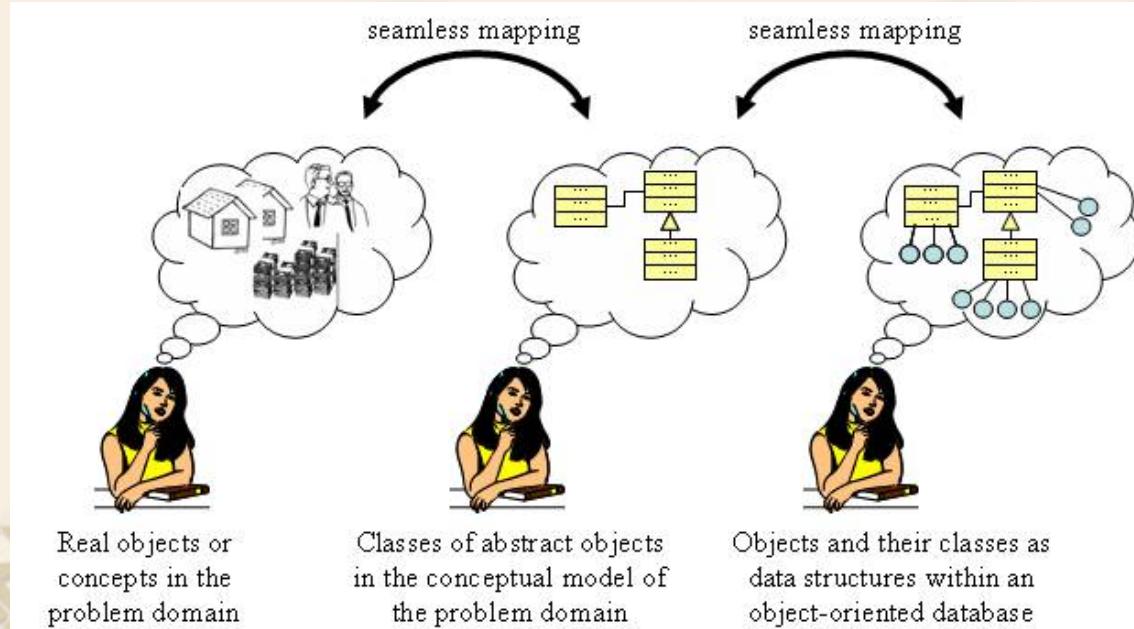


1.4 软件的发展

■ 软件的发展经历了三个阶段

➢ 软件工程阶段 — 80年代以后

开展了有关软件生产技术、软件复用技术、软件生产管理的研究和实践；提出具有广泛应用前景的面向对象方法和相关的语言（Smalltalk、C++）；近年来，软件工程的研究从过程转向产品更加注重程序的开发范型和软件生产。



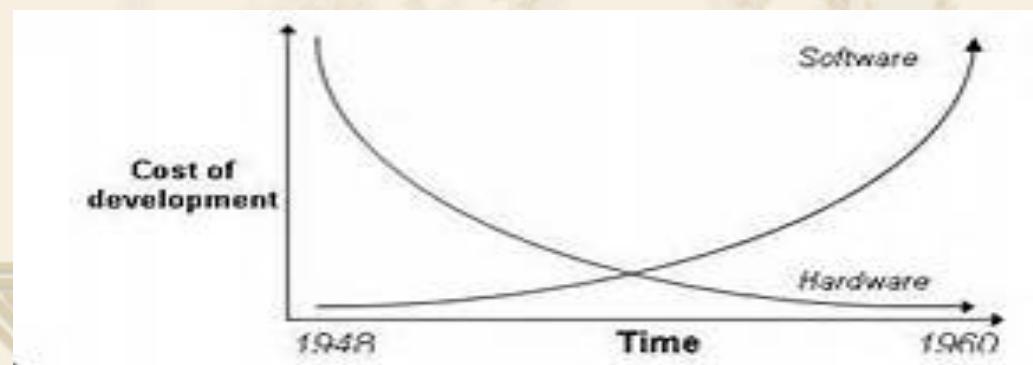
1.5 软件危机

- 对软件开发成本和进度的估算很不准确
- 用户不满意
- 质量不可靠
- 没有适当的文档
- 软件维护困难
- 软件成本比重上升
- 软件开发生产率不能满足人们对合格软件的需求
- 软件工程概念的出现源自软件危机

1.5 软件危机

The major cause of the software crisis is that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem.

— Edsger Dijkstra,



软件危机

危机根源

- 软件本身特点

- 软件本身特点
规模庞大：Windows2000：5000万行代码

- 开发方法的缺陷

- 开发方法的缺陷
忽视需求分析、软件开发 = 程序编写、轻视软件维护

解决途径

- 组织管理

- 组织管理
工程项目管理方法

- 技术措施

- 技术措施
软件开发技术与方法、软件工具

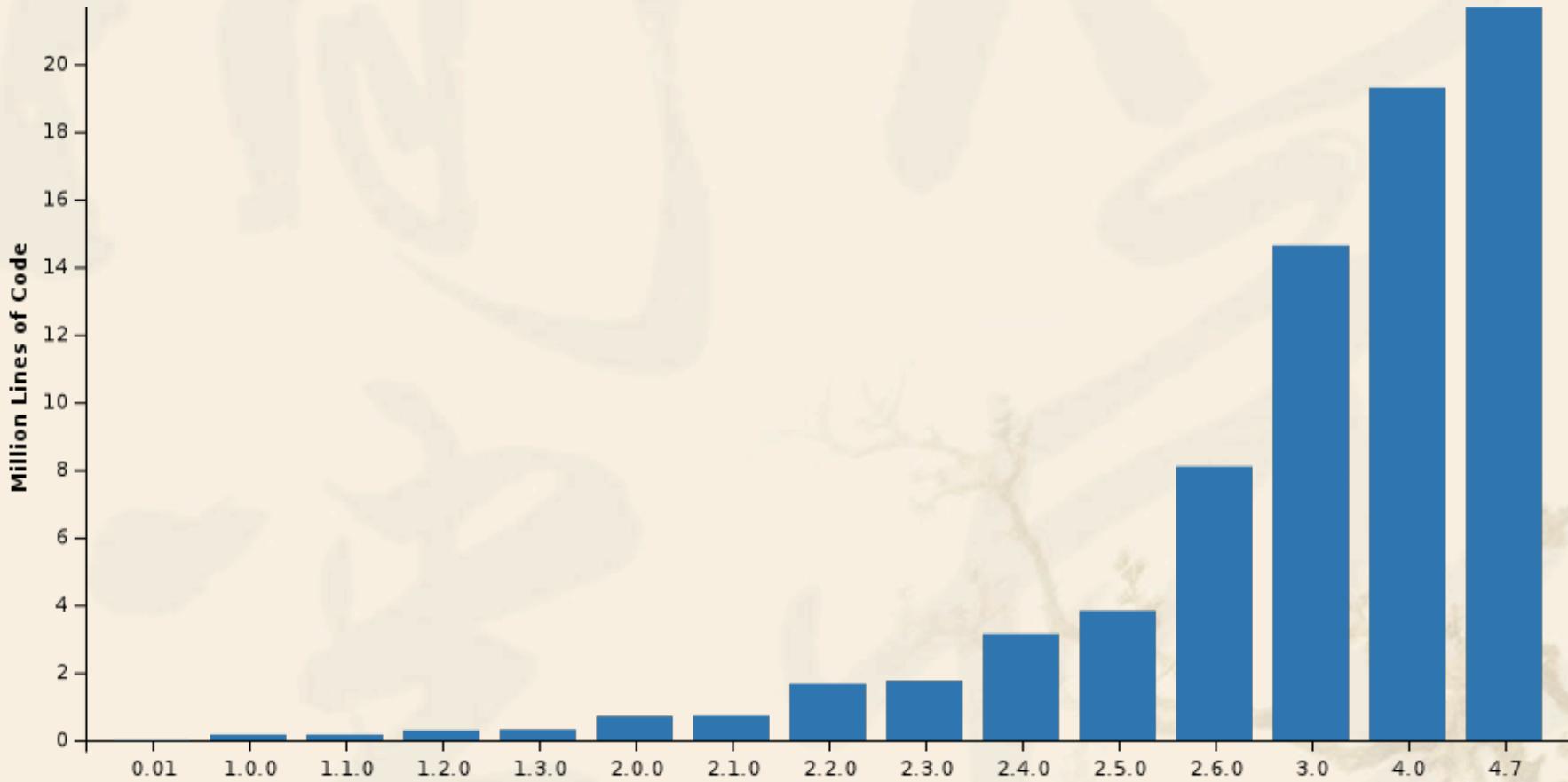
软件危机

❖ Year	Operating System	SLOC (Million)
❖ 1993	Windows NT 3.1	4–5
❖ 1994	Windows NT 3.5	7–8
❖ 1996	Windows NT 4.0	11–12
❖ 2000	Windows 2000	more than 29
❖ 2001	Windows XP	45
❖ 2003	Windows Server 2003	50

软件危机

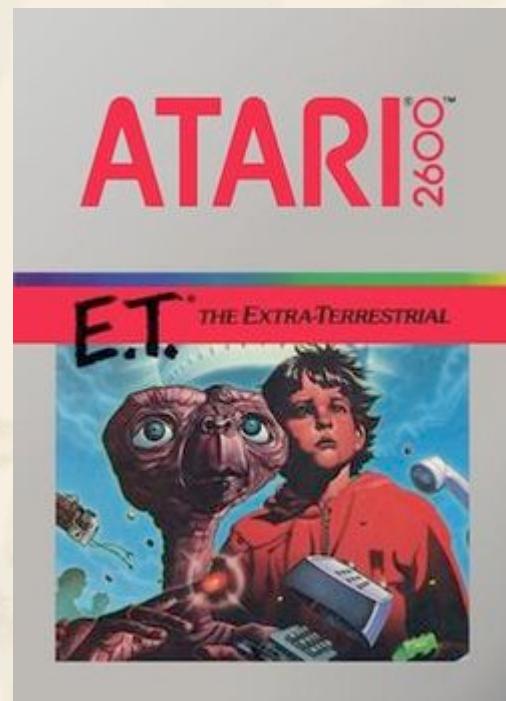
- ❖ 2005 Mac OS X 10.4 86
- ❖ 2001 Linux kernel 2.4.2 2.4
- ❖ 2003 Linux kernel 2.6.0 5.2
- ❖ 2009 Linux kernel 2.6.29 11.0
- ❖ 2009 Linux kernel 2.6.32 12.6
- ❖ 2010 Linux kernel 2.6.35 13.5
- ❖ 2012 Linux kernel 3.6 15.9
- ❖ 2015-06-30 Linux kernel pre-4.220.2

软件危机



软件危机

- ❖ E.T. the Extra-Terrestrial (video game)
- ❖ Atari giving Warshaw **only five and a half weeks** to develop the game in time for the 1982 Christmas season.
- ❖ The result is often cited as one of the **worst video games released** and was **one of the biggest commercial failures in video gaming history**. The game's commercial failure and resulting effects on Atari are frequently cited as a **contributing factor to the video game industry crash of 1983**.



二 软件工程

2.1 软件工程

- 所有发达国家的经济都是依赖于软件的
- 越来越多的系统是由软件控制的
- 在软件工程方面的开支在发达国家 GNP 中占有很大比例
- 软件工程致力于研究低成本的软件开发

Table 2. Value-Added Produced by the Software Industry as a Share of GDP, 1997-2012¹²

Sub-Industry	1997	2000	2005	2010	2012
Computer systems design and related services	0.9%	1.1%	1.0%	1.3%	1.4%
Software publishers	0.5%	0.6%	0.6%	0.7%	0.7%
Data processing, hosting, and information services	0.4%	0.2%	0.5%	0.5%	0.5%
Total	1.7%	2.0%	2.2%	2.4%	2.6%

Table 3. Software Industry Employment, 1990-2014 (thousands)¹³

Sub-Industry	1990	1995	2000	2005	2010	2014
Computer systems design and related services	410	611	1,254	1,195	1,449	1,733
Software publishers	98	157	261	238	261	299
Data processing, hosting, and information services	271	314	473	380	385	471
Total	778	1,083	1,988	1,813	2,095	2,503

Table 4. Software Industry Employment as a Share of Total Private Employment, 1990-2014¹⁴

Sub-Industry	1990	1995	2000	2005	2010	2014
Computer systems design and related services	0.4%	0.6%	1.1%	1.1%	1.3%	1.5%
Software publishers	0.1%	0.2%	0.2%	0.2%	0.2%	0.3%
Data processing, hosting, and information services	0.3%	0.3%	0.4%	0.3%	0.4%	0.4%
Total	0.9%	1.1%	1.8%	1.6%	1.9%	2.2%

Figure 2. U.S. Software Industry Exports

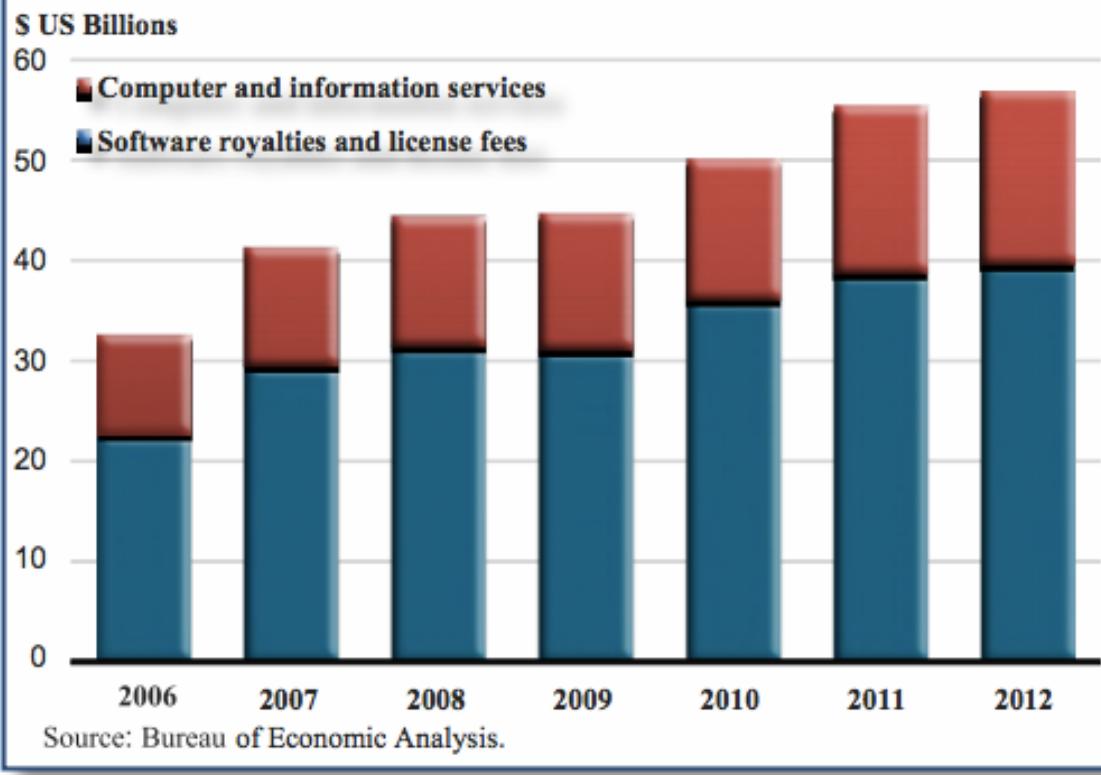
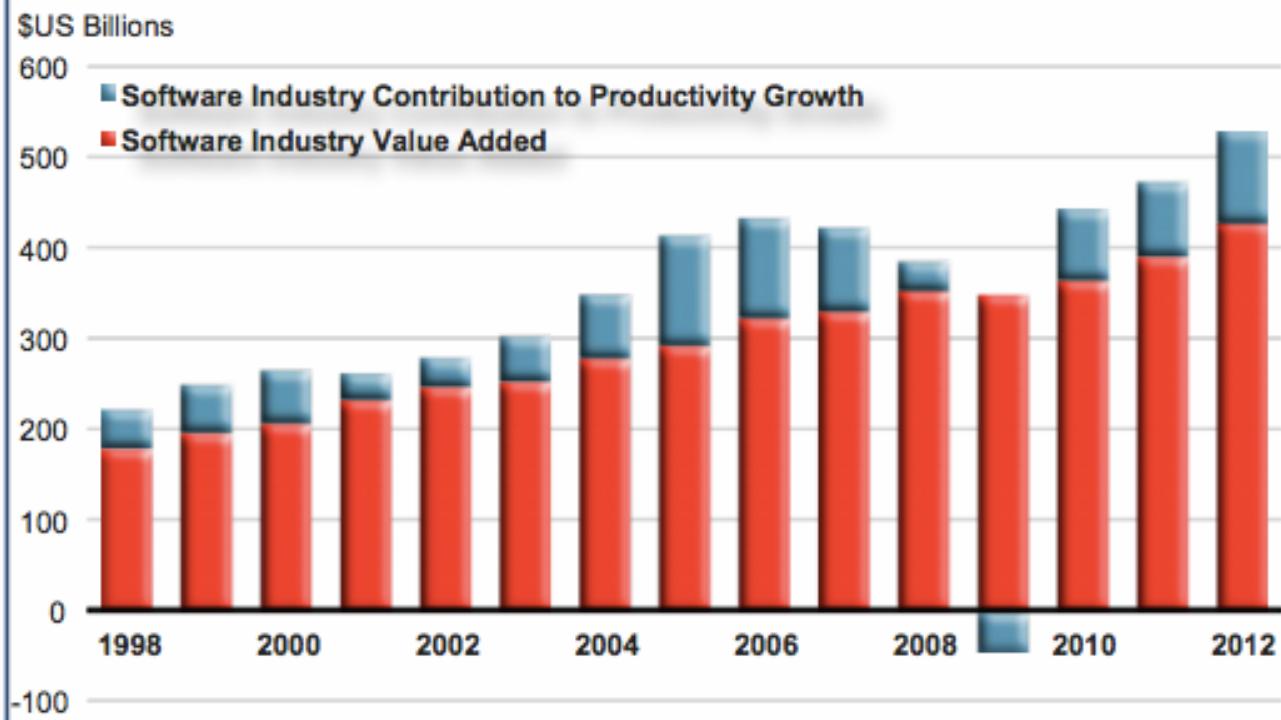


Figure 3. Software's Total Contribution to GDP, 1998-2012



Source: Author's calculations, Bureau

软件工程(*Con.*)

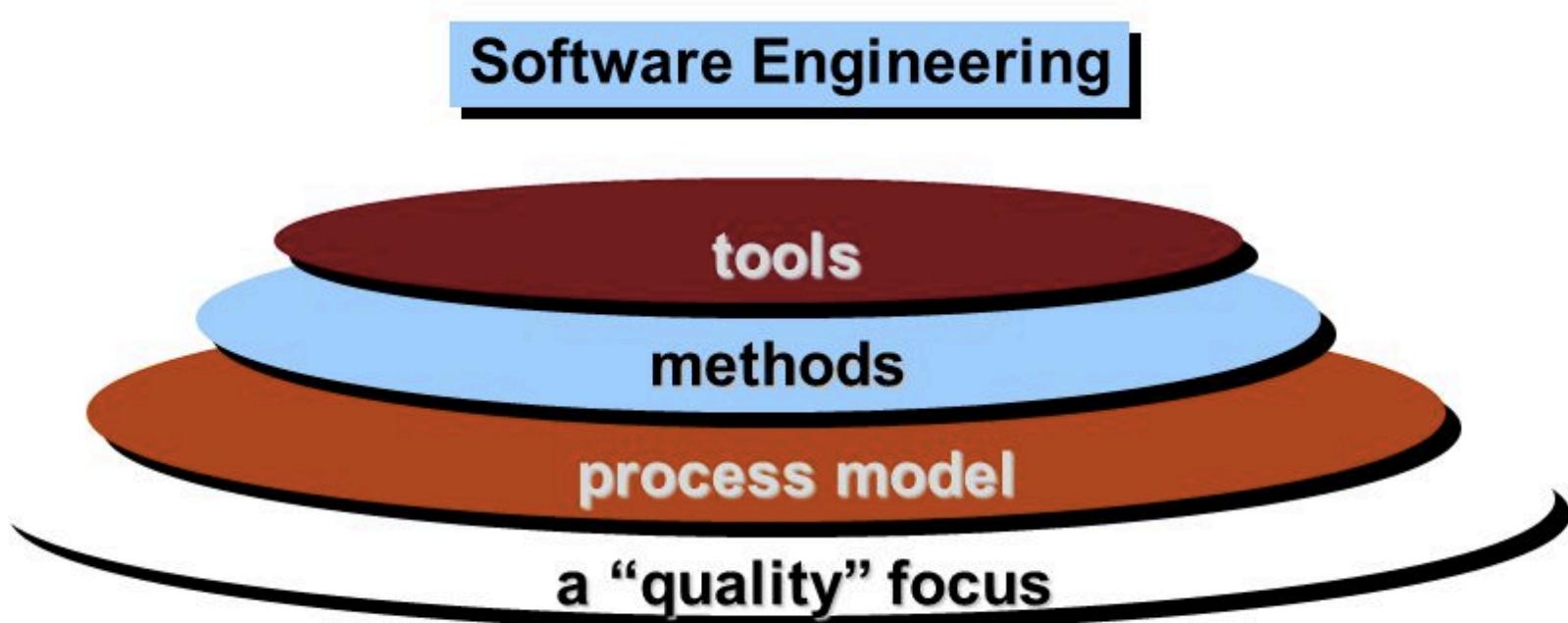
- 软件工程致力于专业化软件开发的理论、方法和工具研究

软件工程方法学的三要素

- 方法Methods: 开发软件的技术，即为软件开发提供了“如何做”的技术
- 工具Tools: 在方法中使用的自动化或半自动化支撑环境，即为软件工程方法提供了自动的或半自动的软件支撑环境
- 过程Process: 获得高质量软件所需要的一系列活动、约束和资源。

Layered View of SE

- Layered approach:
 - Process.
 - Methods.
 - Tools.



2.2 什么是软件工程?

- 软件工程是工程学科，涉及到软件生产的各个方面
- 软件工程师应该有系统的和科学的工作方法，根据要解决的问题，开发上的限制和可利用的资源等因素，选择使用适当的工具和技术

软件工程是应用一组规范化的方法,采用工程的概念、原理、技术和方法,把经过时间考验证明的正确技术和当前能够得到的先进技术和方法结合起来开发、维护和管理软件

2.3 软件工程的特点

- 规范化: 多人共同开发和维护软件时,人人都必须遵循相同的约束规范.
- 文档化: 在软件开发的各阶段都要有相关的文档,用于描述设计思想、设计过程和具体实现技术等信息.
- 工程化: 按照工程化的思想开发和维护软件产品.

2.4 软件工程的本质

- 关注于大型软件产品的构造
- 控制软件的复杂性
- 提高开发软件的效率
- 在开发软件过程中合作
- 满足用户的需求
- 由具有一种文化背景的人替具有另一种文化背景的人创造产品

2.5 软件工程的目标

- 软件工程需要达到的基本目标：
 - 付出较低的开发成本
 - 达到要求的软件功能
 - 取得较好的软件性能
 - 开发的软件易于移植
 - 需要较低的维护费用
 - 能按时完成开发，及时交付使用

2.6 软件工程的基本原理

著名软件工程专家B.W.Boehm综合软件专家、学者们给出出的关于软件开发的意见,并结合其多年开发软件的经验,于1983年在一篇论文中提出了软件工程的7条基本原则

(1) 用分阶段的生命周期计划严格管理

将软件生命周期划分为若干个阶段, 制订实施计划并根据计划进行严格管理

(2) 坚持进行阶段评审

在每个阶段完成时都要进行严格的评审;尽早发现错误, 尽量避免将错误带入下一阶段

软件工程的基本原理

(3) 实行严格的产品版本控制

基线配置：经过阶段评审后的软件配置成分（文档、源代码）。实行基线配置管理，不得随意变动。

(4) 采用现代程序设计技术

- 采用先进的技术（结构化程序设计技术、面向对象程序设计技术、构件程序设计技术），提高软件开发的效率和软件产品的质量。

软件工程的基本原理

(5) 结果应能清楚地审查(文档支持)

- 软件开发的管理和软件产品的结果应有利于审查。

(6) 开发人员应该少而精

- 开发组人员素质高;
- 人数越少越好.

(7) 承认不断改进软件工程实践的必要性

- 积极主动地采纳新的软件技术;
- 不断总结经验.

三 软件工程中的常见问题

软件工程中的常见问题

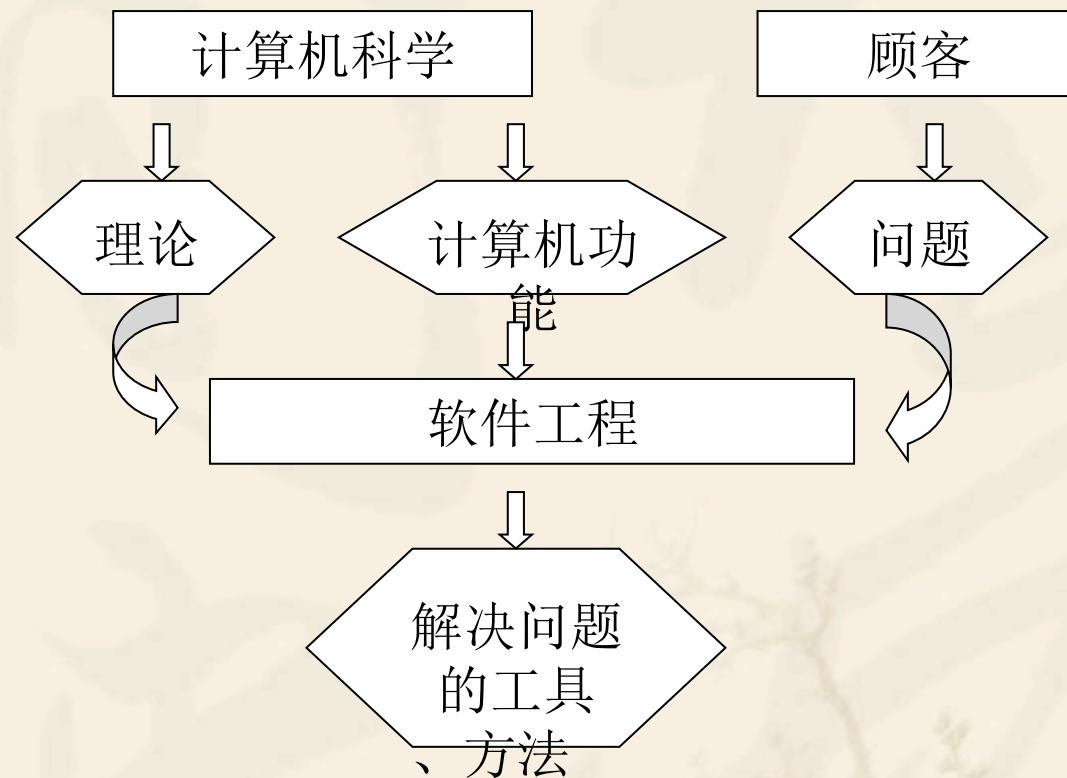
- 软件工程和计算机科学的区别？
- 软件工程和系统工程的区别？
- 什么是软件过程？
- 什么是软件过程模型？

软件工程中的常见问题

- 什么是软件工程的成本？
- 什么是软件工程方法？
- 什么是**CASE (Computer-Aided Software Engineering)**？
- 优良软件的特点？
- 软件工程面临的主要挑战？
- 有什么百科全书式的关于软件工程的资料？

3.1 软件工程和计算机科学的区别？

- 计算机科学研究的是构成计算机和软件系统的基础的有关理论和方法，而软件工程研究软件生产中的实际问题
- 计算机科学理论现在已经不够做为软件工程学的完全支撑



计算机科学和软件工程间的
关系

3.1 软件工程和计算机科学的区别？

- 专业，职业教育中有软件工程专业。
- 计算机科学在本科教育中。

3.2 软件工程和系统工程的区别？

- 系统工程是基于计算机的系统工程，包括硬件开发、软件开发、过程设计、系统实施等。软件工程是系统工程的一部分
- 系统工程师需要负责系统描述、体系结构设计、整合和部署

Systems Engineering

Human
Factors
Engineering

Software
Engineering

Verification
Engineering

Hardware
Engineering

Requirements Engineering

Soft Systems
Engineering

Enterprise
Management

Project
Management

3.3 什么是软件过程？

- 软件过程是生产软件产品的一系列活动及其结果
- 软件过程包含的4项基本活动：
 - 软件描述 - **what the system should do and its development constraints**
 - 软件开发 - **production of the software system**
 - 软件有效性验证 - **checking that the software is what the customer wants**
 - 软件进化 - **changing the software in response to changing demands**

3.4 什么是软件过程模型？

- 软件过程模型是从一特定的角度提出的软件过程的简化描述
- 一般的软件过程模型
 - **Waterfall**
 - **Evolutionary development**
 - **Formal transformation**
 - **Integration from reusable components**

3.5 软件工程的成本包括哪些？

- **60%**的成本是开发成本, **40%** 是测试成本. 对于定制软件, 软件进化成本会超过开发成本
- 软件开发成本是依赖于系统的开发方法、系统特性的要求
- 软件成本的分布依赖于所采用的开发模型

Developer vs Tester



I am not able to replicate this issue. This is working fine on my machine. So close this bug.

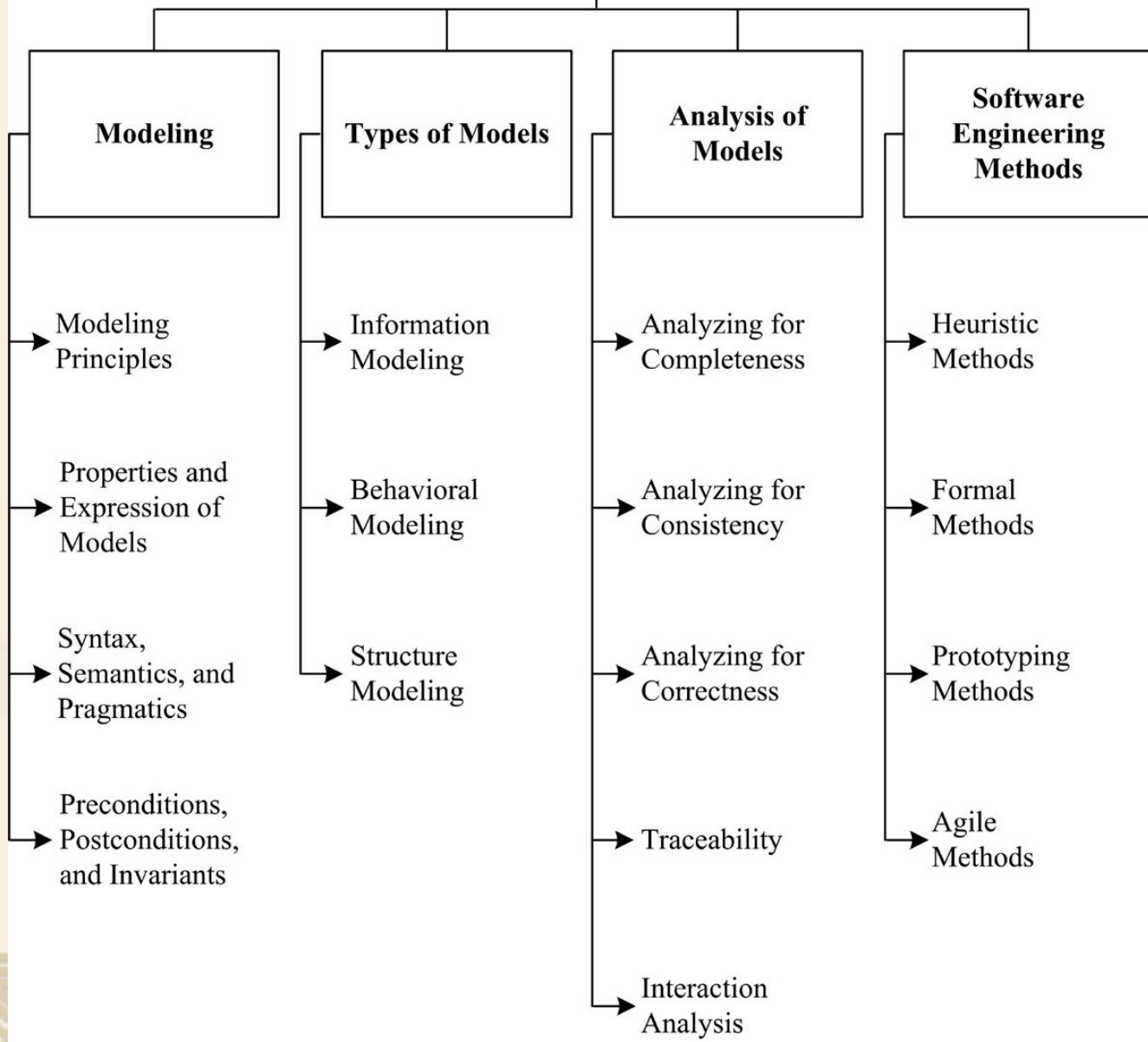


I don't care if it is working fine on your machine. We are not going to deliver your machine to Client.

3.6 什么是软件工程方法？

- 软件工程方法是一种软件开发的结构化方法，其目的是使高质量的软件生产性价比比较高

Software Engineering Models and Methods



3.7 什么是CASE (Computer-Aided Software Engineering)

- **CASE**是计算机辅助软件工程，它包括支持软件过程活动（需求分析、系统建模、调试和测试等）的软件工具
- 高端**CASE**工具
 - Tools to support the early process activities of requirements and design**
- 低端**CASE**工具
 - Tools to support later activities such as programming, debugging and testing**

3.8 软件包括哪些属性?

- 软件除了提供相应的功能外，作为一个产品它还有一系列相关的反映质量的属性
- 可维护性
 - Software must evolve to meet changing needs**
- 可依赖性
 - Software must be trustworthy**
- 有效性
 - Software should not make wasteful use of system resources**
- 可用性
 - Software must be usable by the users for which it was designed**

3.9 软件工程面临的主要问题？

- 遗留系统的挑战
Old, valuable systems must be maintained and updated
- 异构性的挑战
Systems are distributed and include a mix of hardware and software
- 交付上的挑战
There is increasing pressure for faster delivery of software

3.10 软件开发的参与者

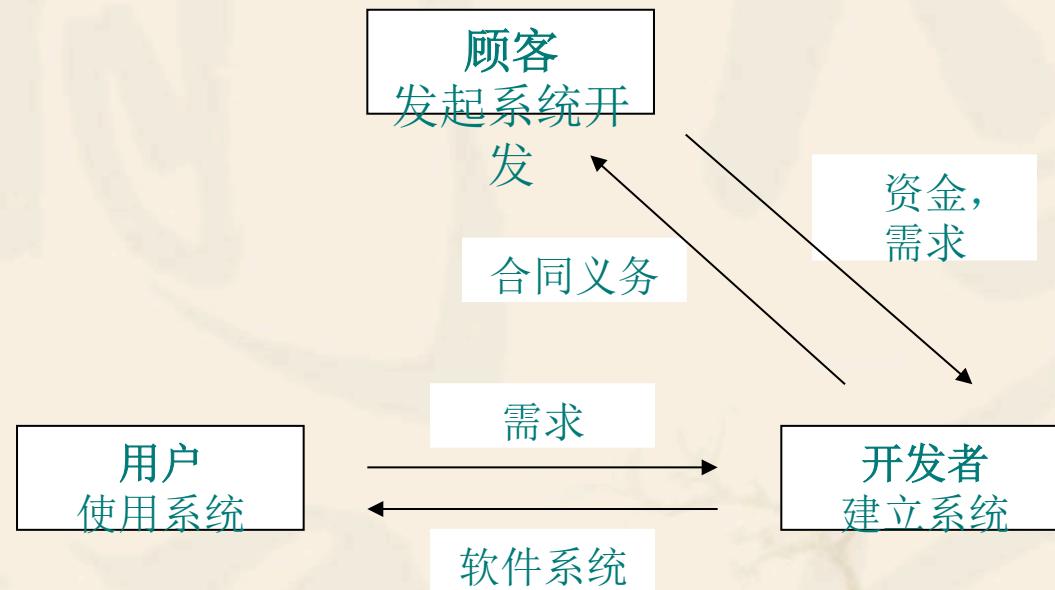


图1.7 软件开发的参与者

有什么百科全书式的关于软件工程的资料？

About SWEBOK

The *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)* describes generally accepted knowledge about software engineering. Its 15 knowledge areas (KAs) summarize basic concepts and include a reference list pointing to more detailed information. For SWEBOK Guide V3, SWEBOK editors received and replied to comments from approximately 150 reviewers in 33 countries.

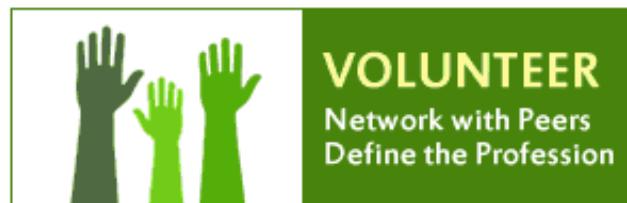
A .PDF version of the Guide is available free to all through the IEEE Computer Society.

The SWEBOK Guide has also gained international recognition as ISO Technical Report 19759.

In future refreshes, the Computer Society and its volunteers will continue to use the transparent and open consensus process that is an integral part of SWEBOK.



Click [here](#) to download the SWEBOK logo.



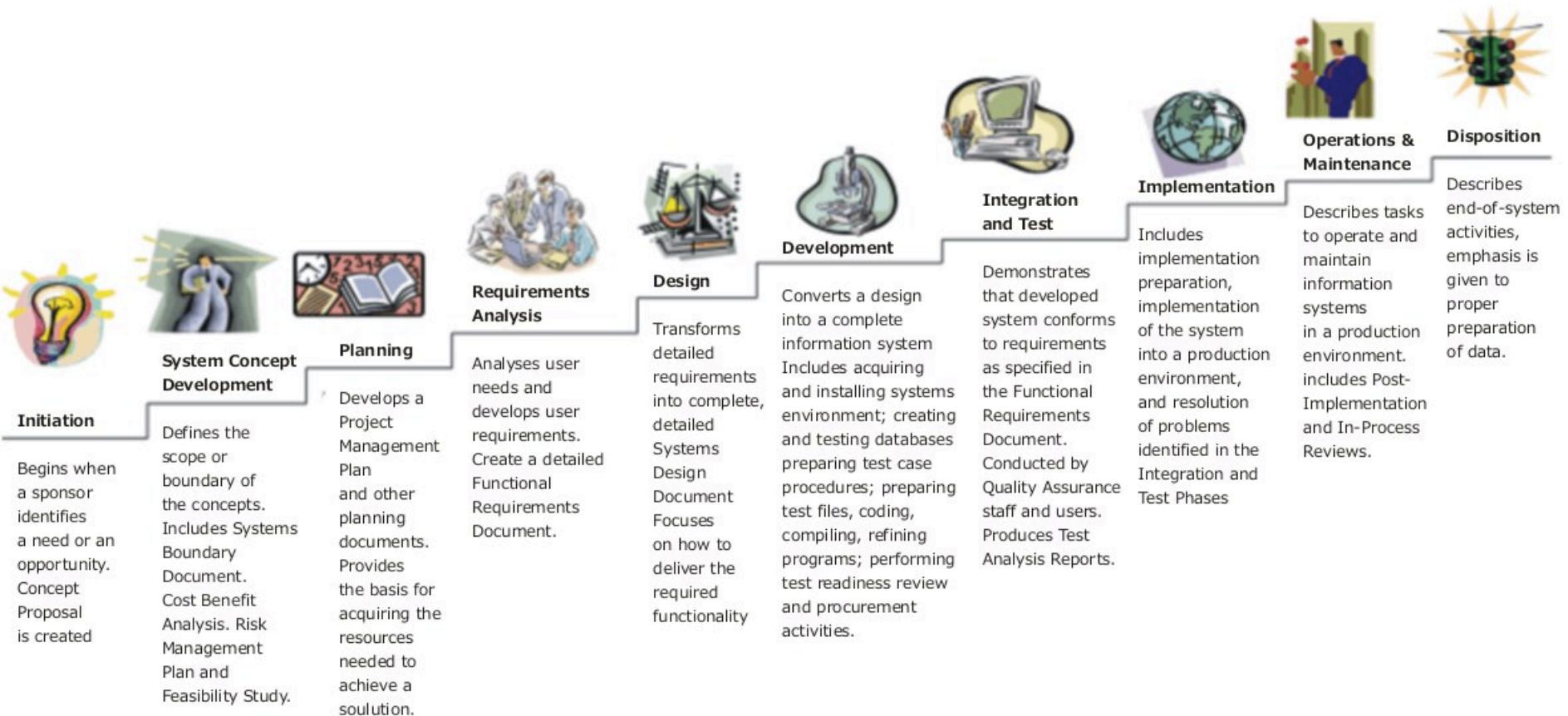
四 软件生存周期

4.1 软件生存周期

- 软件有一个孕育、诞生、成长、成熟、衰亡的生存过程。这个过程即为软件的生存期
- 生命周期(life cycle): 当过程涉及到建造某种产品时,常将过程称之为“生命周期”.
- 软件生命周期SLC(Software Life Cycle): 软件开发和维护的过程称之为软件生命周期,它描述了软件产品从设想到实现、提交、使用和维护的过程.

Systems Development Life Cycle (SDLC)

Life-Cycle Phases



软件生存周期

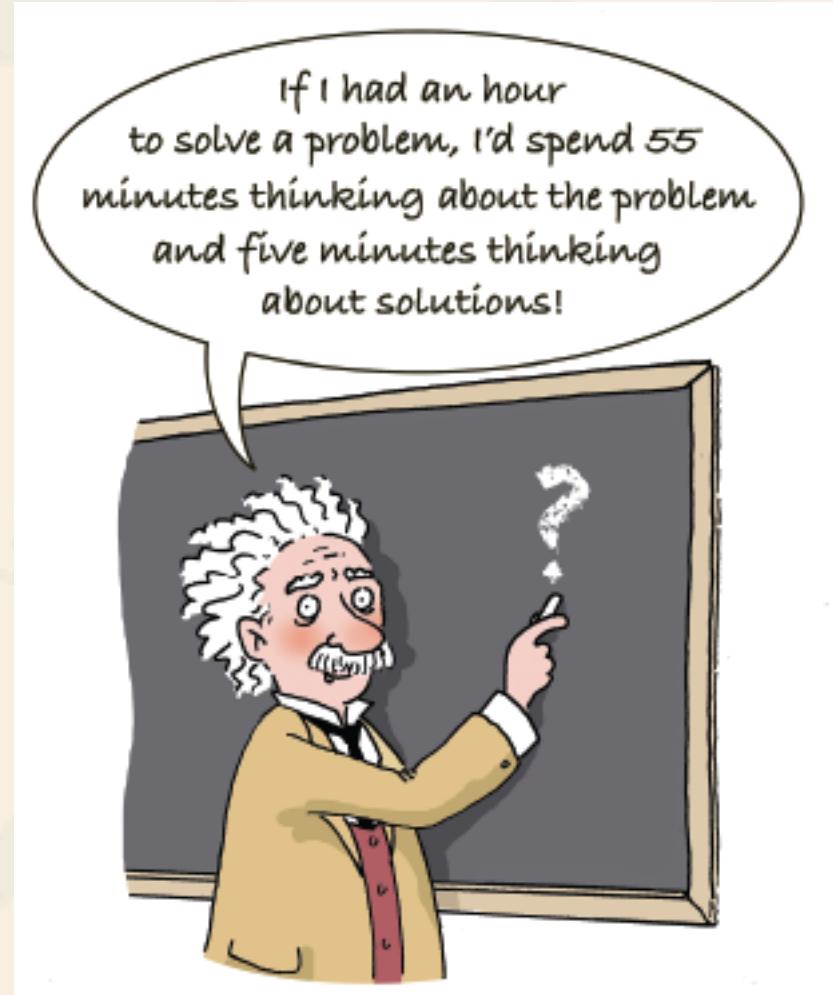
软件生存期六个步骤：

- 制定计划(**Planning**)：问题定义与可行性研究
- 需求分析和定义(**Requirement Analysis and Definition**)
- 软件设计(**Software Design**)
- 程序编写(**Coding, Programming**)
- 软件测试(**Testing**)
- 运行/维护(**Running/Maintenance**)

(1) 问题定义

问题定义

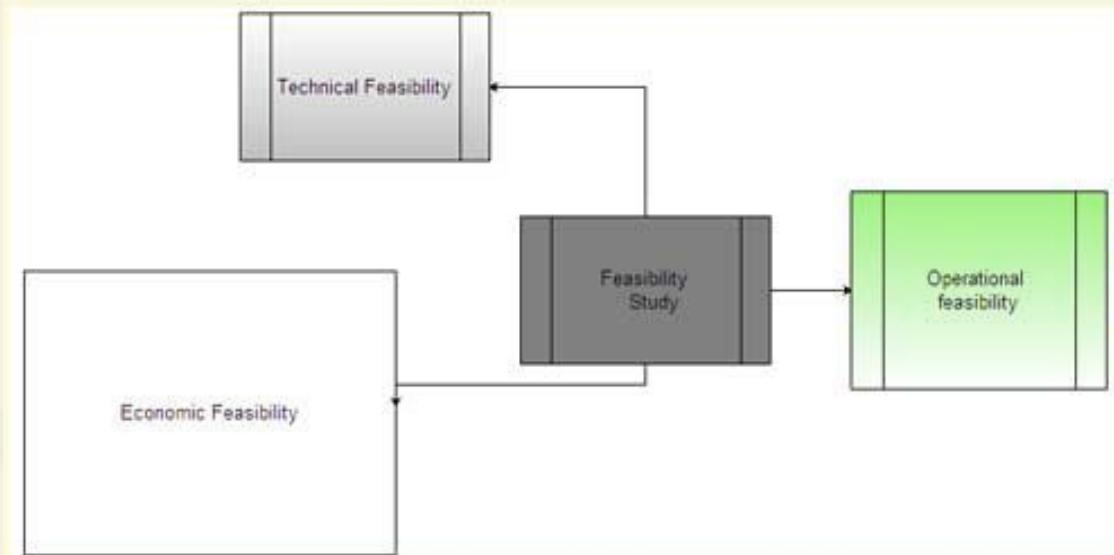
- 问题是什么
- 确定软件开发工程的总体目标
- 确定问题的性质、项目规模和系统边界
- 提交《问题定义报告》



(2) 可行性研究

可行性研究

- 有可行解吗？
- 成本/效益分析
- 确定工程的可行性
- 提交《可行性研究报告》
- 对《可行性研究报告》进行评审



(3) 项目计划

项目计划

- 估计可利用的资源(计算机硬件， 软件， 人力等)、 成本、 效益、 开发进度。制定《软件项目开发计划》
- 制定出完成开发任务的实施计划,连同可行性研究报告， 提交管理部门审查。对《软件项目开发计划》 进行评审

(4) 需求分析

需求分析

- 获取用户的需求,明确用户要求系统做什么 , 做到什么程度; 确定系统的功能需求、性能需要和其他需求
- 编写《用户需求报告》
- 根据《用户需求报告》编写《需求规格说明书》
- 对《用户需求报告》和《需求规格说明书》进 行评审

(5) 总体(概要)设计

总体(概要)设计

- 确定系统实现方案
- 对各种方案进行评价,选择一个合适的方案
- 系统流程图 + 层次图或结构图
- 提交《总体设计报告》
- 对《总体设计报告》进行评审

(6) 详细设计

详细设计

- 设计每个模块内部结构
- 设计数据结构和模块内部的算法
- 提交《详细设计说明书》
- 对《详细设计说明书》进行评审

(7) 编码和单元测试

编码和单元测试

- 编写程序代码
- 进行单元测试
- 提交源程序清单和目标程序
- 对源程序进行评审
 - Code+测试方案和结果

(8) 测试

- **系统(集成,综合)测试**

- 集成测试：将经过单元测试的模块组装，并进行部件测试。
- 功能测试：测试集成后系统的功能。
- 性能测试：测试系统的性能。
- 安装测试：在用户环境进行测试。
- 验收测试：验收通过后，由用户签字确认。
 - 符合要求的软件

(9) 产品发布

软件产品提交

- 完善软件文档
- 用户培训
- 软件产品包装
- 软件产品发布

(10) 维护

运行 / 维护

- 改正性维护 运行中发现了软件中的错误需要修正
- 适应性维护 为了适应变化了的软件工作环境，需做适当变更
- 完善性维护 为了增强软件的功能需做变更

持久地满足用户需要

五 软件工程人员的职业道德

5.1 职业和道德上的责任

- 软件工程人员必须承认他们的工作不仅仅是技术的应用，还要担负许多责任
- 软件工程人员要受人尊敬，其行为必须符合道德，必须有责任心
- 某些行为没有法律加以规范，只能靠职业道德来约束

5.2 软件工程人员的职业道德

- 机密

工程师应该尊重他们雇主和客户的机密，无论是否签署过正式的机密协议

- 工作能力

工程师不应该夸大自己的工作能力。不应该故意接受超出自己能力的工作

5.2 软件工程人员的职业道德(Con.)

- 知识产权

工程师应该知道地方性的有关知识产权的使用方面的法律，例如专利权，版权等。应该认真履行职责确保顾主和客户的知识产权不受侵害。

- 计算机滥用

软件工程师不应该在他人的机器上滥用他们的技术。计算机的滥用包括从不起眼的小事（在老板机器上打游戏，等）到极端严重的事件（发布病毒）。

5.3 ACM/IEEE 职业道德规范

- ACM和IEEE推出了关于职业道德职业行为的规范
- 规范包含与行为和意识相关的八项原则，包括从业者、教育家、经理、管理者和政策制定人，也包括职业的实习生和学生

5.3.1 道德规范 - 序言

序言

- 规范的简化版本在较高的层次上提出了要求；在完整的版本中条款举例说明了这些要求如何改变我们作为 软件工程专业人士的生存方式。没有要求，这些细节就编成了生硬和琐细的法律条文； 缺少细节，要求只能是高调和空谈； 二者的结合使之成为完整的规范
- 软件工程师应当做出承诺，使软件的分析、描述、设计、开发、测试和维护等工作对社会有益且受人尊敬。 基于对公众健康、安全和福利的考虑，软件工程人员应当遵守以下八条原则：

5.3.2 道德规范 – 八条原则

1. 公众感

- 软件工程师应该始终与 公众的利益保持一致。

2. 客户和雇主

- 软件工程师在公众利益的前提下满足客户和雇主的最大利益。

3. 产品

- 软件工程师应当保证他们的产品和相关的附件达到尽可能高的行业标准。

5.3.2 道德规范 - 八条原则(Con.)

4. 判断力

- 软件工程师应该具有公正和独立的判断能力。

5. 管理

- 软件工程经理和领导者应该拥护并倡导合乎道德的有关软件开发和维护的管理方法。

6. 职业感

- 软件工程师应当弘扬职业正义感，尊重社会公众利益。

5.3.2 道德规范 - 八条原则(Con.)

7. 同事

- 软件工程师应该公平地对待和协助他们的同事。

8. 自己

- 软件工程师应当毕生学习专业知识，倡导合乎职业道德的职业生活方式。

要点

- 软件工程是一门工程学科，涉及软件生产的各个方面
- 软件产品由开发的程序、相关文档及数据构成。软件产品的基本属性是可维护性、可依赖性、有效性、可用性
- 软件过程由开发软件产品的一系列活动组成。基本活动有：
：软件描述、开发、有效性验证和进化
- 软件工程方法是软件生产的组织方式，包括软件过程的建议、使用的标记法、进行系统描述的规律和设计指南

要点

- **CASE**工具是一些软件系统，被设计成支持软件过程中的常规活动、如编辑设计图表、检查图表的连贯性、跟踪已经运行的程序测试等
- 软件工程人员对软件工程这一职业和社会负有责任，不应该只关心技术问题
- 职业协会颁布的行为准则规定了一系列协会成员应该遵守的行为准则