

高级软件工程

chapter 1 Introduction

衣杨

issyy@mail.sysu.edu.cn ; 815583456@qq.com

Wechart: [yiyangSYSU](#)

本课程的安排

■ 理论

- 内容：基本原理、方法和技术
- 形式：讲授、课堂讨论、作业、作业展示

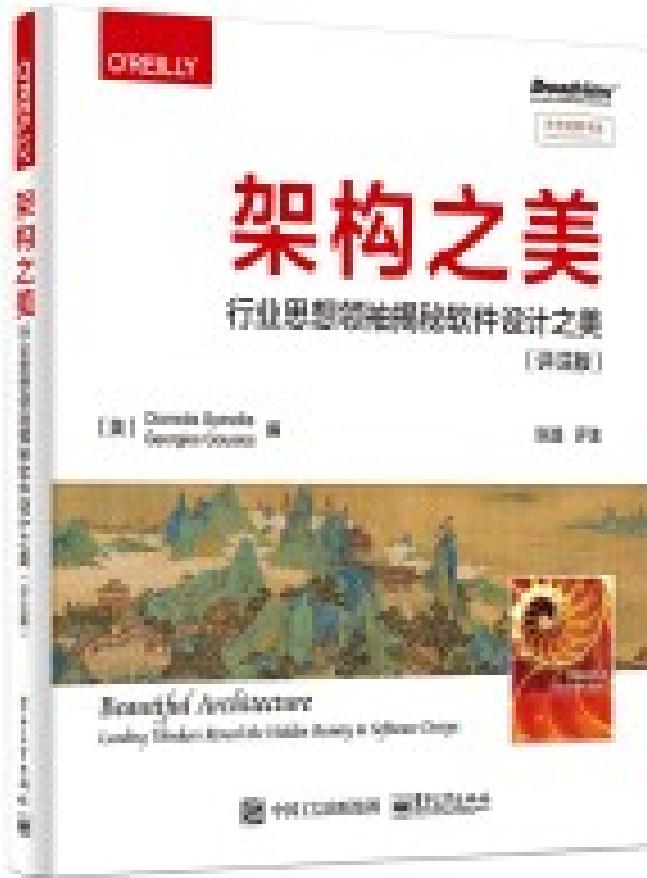
■ 实践

- 内容：构造一个应用系统（包括分析、设计、编码、测试）
- 形式：自己完成、分阶段文档、编码、提交

推荐几本书

1. 齐治昌，谭庆平，宁洪，软件工程（第3版），高等教育出版社，2012
2. 钱乐秋、赵文耘、牛军钰，软件工程（第3版），清华大学出版社，2016
3. 罗杰 S. 普莱斯曼，布鲁斯 R. 马克西姆，郑人杰等（译），软件工程：实践者的研究方法（原书第8版），
4. JAVA/C++/PHP/MY SQL
5. UML 资料



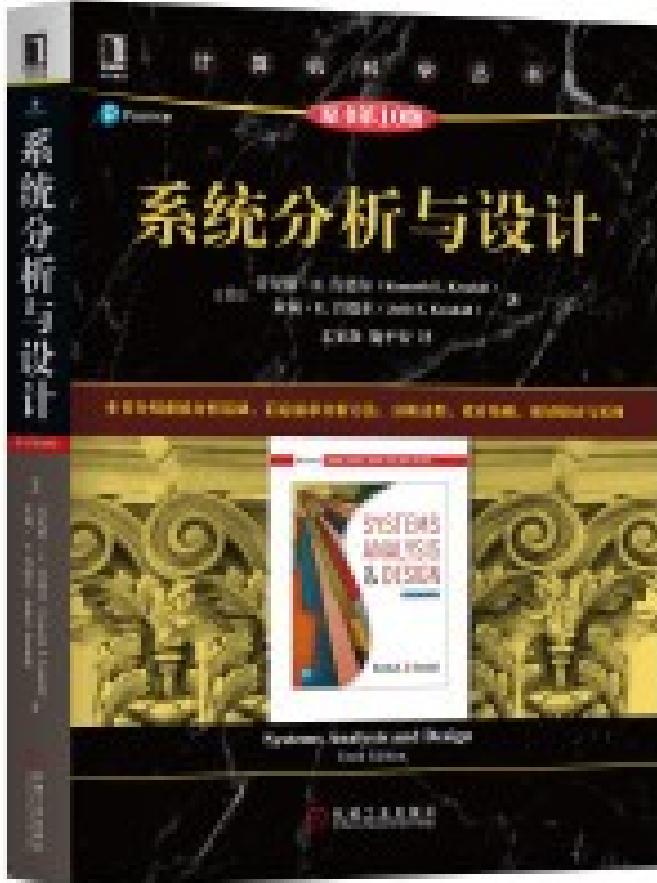


架构之美：行业思想领袖揭秘软件设计之美（评注版）

跟业内专家一起领略软件架构设计中的简洁之美、清晰之美、风格之美、灵活之美和演进之美。

作者：(美) Diomidis Spinellis (迪奥米德斯·斯宾耐立思) , Georgios Gousios (乔治斯·郭西奥斯)

出版社：电子工业出版社
出版时间：2018年06月



系统分析与设计（原书第 10 版）

本书分别介绍了系统分析基础、信息需求分析以及分析过程、设计基础和质量保证与实现。

作者：【美】肯尼斯·肯德尔
(Kenneth E. Kendall) 【美】朱莉·肯

出版社：机械工业出版社
出版时间 :2019 年 12 月

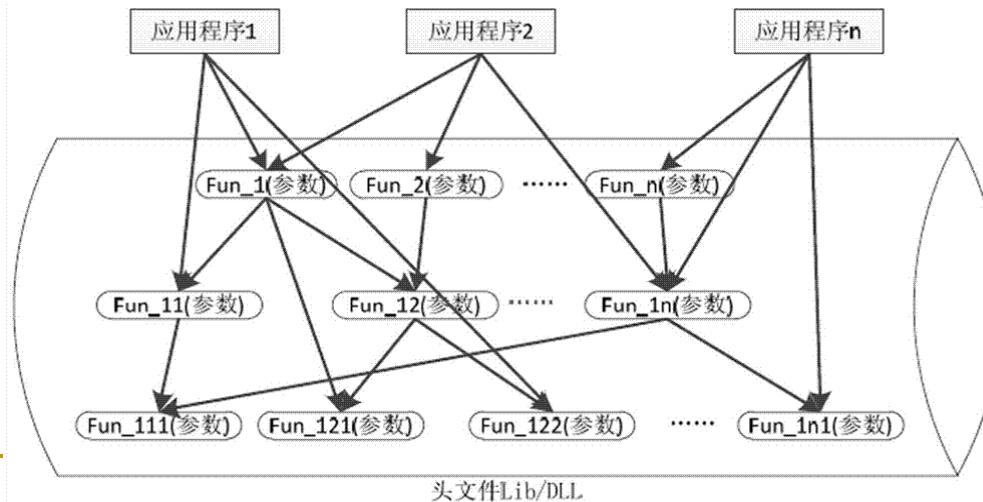
软件 = 程序 + 算法？

- 用 C 实现了二叉树遍历，这个程序有什么实际用途？用 Java 实现了，似乎没有指针，那么可以不必了解二叉树了吗？



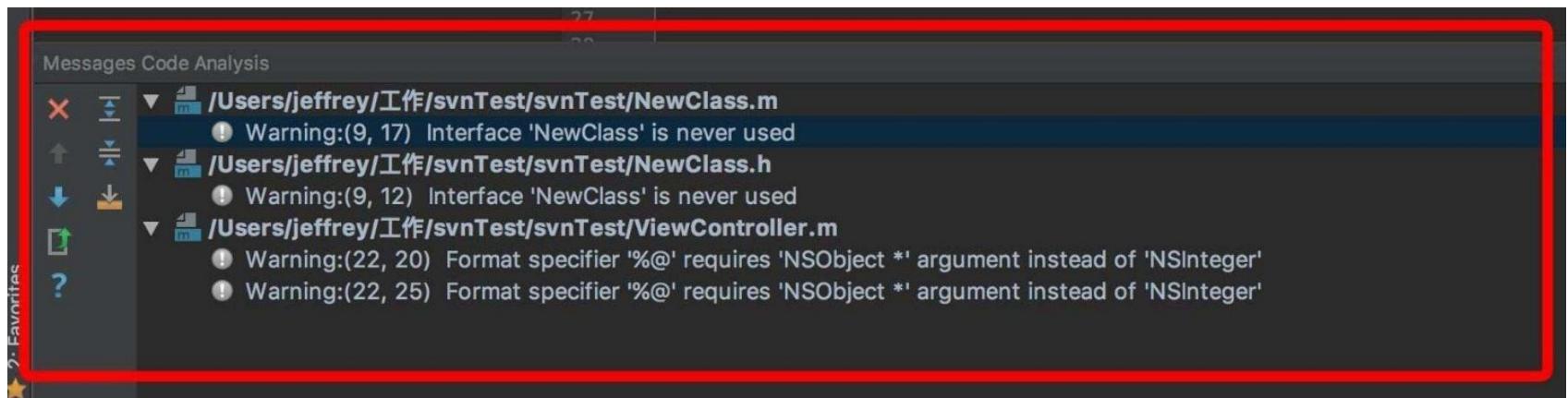
软件 = 程序 + 算法？

- 似乎所有算法都被别人实现了，我调用即可，学习数据结构和算法有啥用？
- 如何区分一个好的程序员和不好的？



软件 = 程序 + 算法？

- 入职后，发现以前同事写的代码是垃圾，愤怒的要重写，一个老员工说，这是去年新来的程序员推翻之前重写的，但是感觉还不如之前的好，



■ 软件行业赖以生存的软件，程序员安 生立命的程序，到底是什么？

移山公司程序员阿超辅导儿子

- 老师要求每天孩子出 30 道加减法题目

()-0=6	10-()=8	4+()=9	7-()=6	()-3=0
()+7=8	5-()=2	()-5=5	()+6=9	1+()=8
7-()=7	6+()=10	()+2=8	()-3=4	3+()=4
9-()=0	()+6=7	4+()=8	()-9=1	()-3=5
()+1=4	()-7=4	()+8=10	9-()=4	()-5=1
4+()=10	()+5=5	()-2=5	10-()=2	()-6=4
()-4=2	7-()=2	10-()=1	()+3=8	5+()=7
()+3=9	2+()=9	()-7=2	()-3=0	6-()=2
3-()=2	10-()=4	()-1=8	10+()=10	()+7=8

- 阿超写了一个程序（估算工作量）

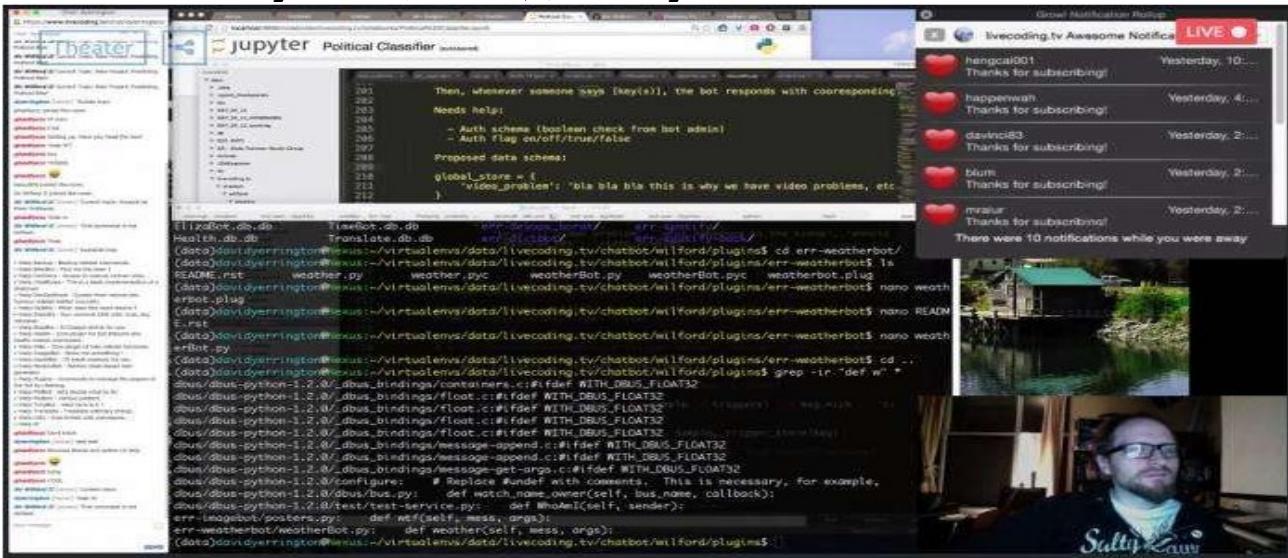


移山公司程序员阿超辅导儿子

- Excel,C/C++,VB,Unix Shell,VB

Scrip,Java,Perl,Python,.....

- 打印出来，给娃做了，老师看到。。。



移山公司程序员阿超辅导儿子

- 老师请阿超做一个，给 2-4 年级学生用



老师的要求

- 题目避免重复
- 可定制数量和打印方式
- 可以控制
 - 是否有乘除法
 - 是否有括弧
 - 数值范围
 - 加减有无负数
 - 除法有无余数
 - 是否支持小数、到第几位、打印中每行的间隔。 . .

```
if ( support_plugin ){
    upload_obj = plugin;
} else if ( support_html5 ){
    upload_obj = html5;
} else if( support_flash ){
    upload_obj = flash;
} else{
    upload_obj = form;
}
```

- 儿子告诉阿超：老师明天要
- 阿超：随手一个程序怎么来了一个用户，还有不少需求？
- 阿超熬夜写好，交给老师。。。。
- 大家估算工作量

- 教导主任看到，要：
- 建议把这个程序放到学校网站
- 增加一点点要求
- 支持二元一次方程，能开根号，可以生存期中、期末试卷，5AM-10PM 可访问
- 大家估算工作量

阿超给儿子辅导作业的过程：

- 程序一
- 软件一
- 服务

■ 软件

- 合理的软件架构 (Software Architecture)
- 软件设计与实现 (Software Design, Implementation and Debug)

■ 源代码管理

- 配置管理 (Software Configuration Management)
- 质量保障 (Quality Assurance)
- 软件测试 (Test)

■ 需求分析 (Requirement Analysis)

- 顾客会提出各种靠谱和不靠谱的要求
- 有些能做到，有些做不到
- 用户体验 (User Experience)
- 国际化和本地化 (Globalization & Localization)

■ 软件项目管理 (Project Management)

□ 软件生命周期 (Software Life Cycle, SLC)

- 程序理解 (Program Comprehension)
- 软件维护 (Software Maintenance)
- 服务运营 (Service Operation)
- 道德规范

团队如何养活自己

- 交钱买断
- 先试用再交钱
- 免费
- 免费并提供源代码
- 买硬送软，买软送硬
- 免费，但是要播广告
- 免费，维护。。。



推广

- 软件 = 程序 + 软件工程
- 软件企业 = 软件 + 商业模式
 - 还包括，人才招聘、绩效评估、升迁、淘汰
- 算法和数据结构是基本功，在此之上，软件工程决定了软件的质量，商业模式影响了一个软件企业的成败

练习与讨论

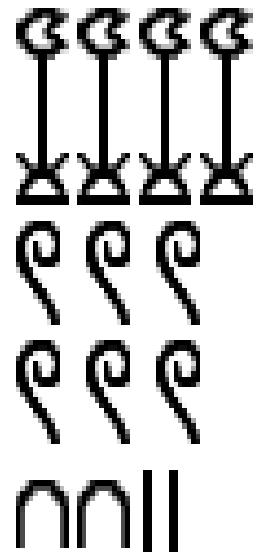
- 科研和创新
- 同学们参看李凯教授的文章：促进中国科技创新的想法。
- 结合自己实际，畅谈你心目中理想的科研和创新

Initialization

Case : 表达的重要 (数字表达)

- 古埃及
- 古罗马 : I, II,
- 玛雅文明 : 二十进制
- 阿拉伯数字 : 1, 2,
- 中国 : 一 , 二 ,

卡纳克神庙中的
石刻 : 4622



Case : 表达的重要（数字表达）

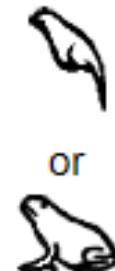
- 古埃及
- 古罗马 : I, II,
- 玛雅文明 : 二十进制
- 阿拉伯数字 : 1, 2,
- 中国 : 一, 二

卡纳克神
庙中的石
刻 : 4622



数值	1	10	100	1,000	10,000	100,000	1 百万, 或指 大量
----	---	----	-----	-------	--------	---------	----------------

象形文字



or



蝌蚪
or 青蛙

Case : 表达的重要（数字表达）

- 古埃及托勒密王朝 --Rosetta Stone
(罗塞塔石碑)
- 计算机用什么表达？为什么？

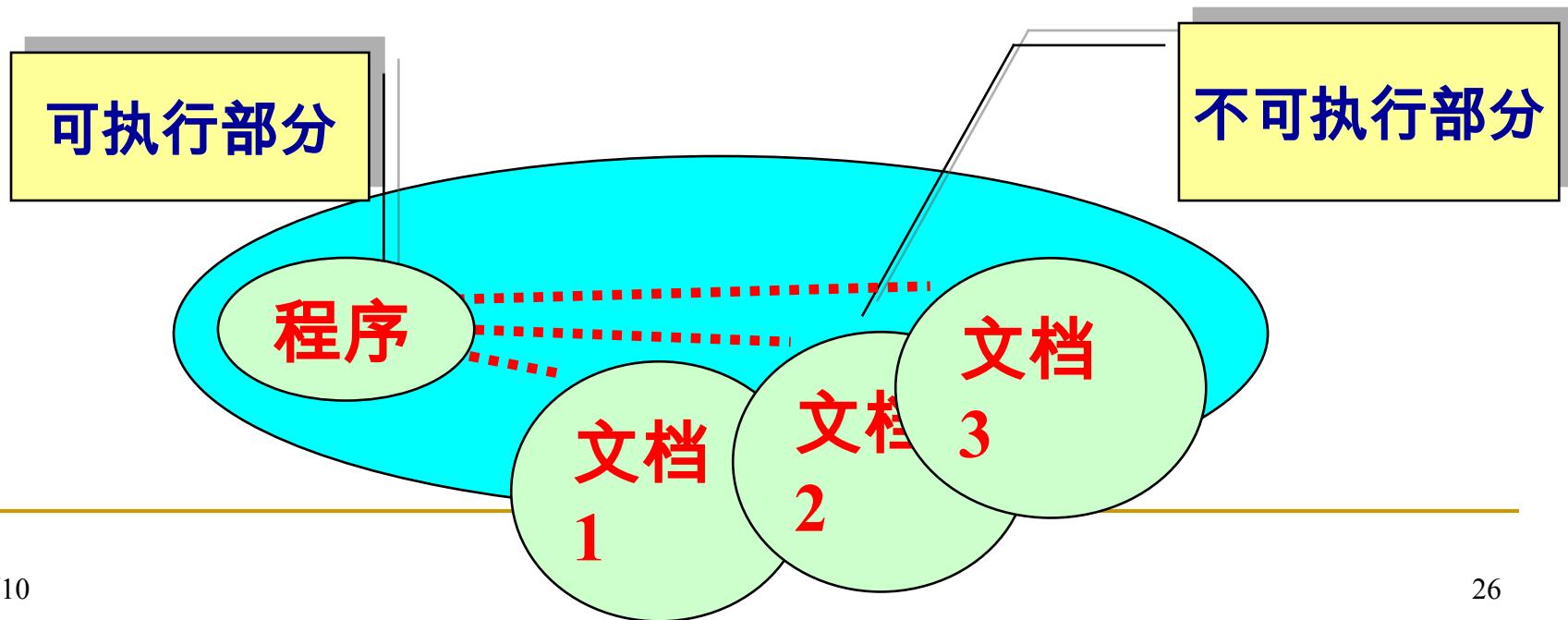


2.1 软件工程背景和意义

什么是计算机软件？



- 与操作有关的**程序**、及与之相关的**文档**集合

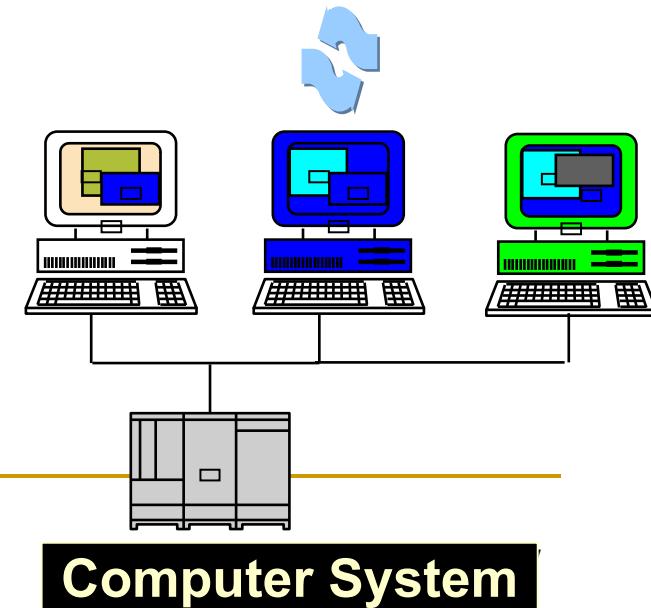
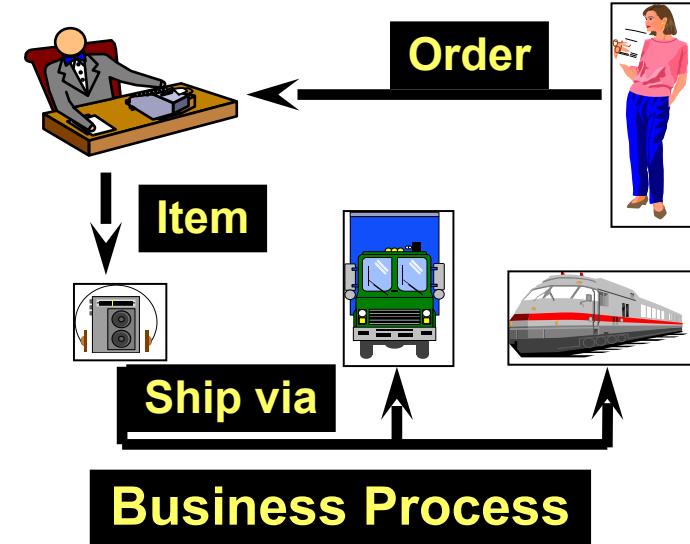


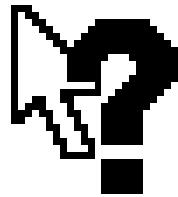
2.2 软件工程背景和意义

什么是程序 (1/2) ?

- 能为计算机识别、理解和处理的语句序列
- 程序例子

```
Main()  
{ int i, j;          // 变量定义  
    char Str[10];  
    i = i + j ;      // 语句说明  
    ..... }
```

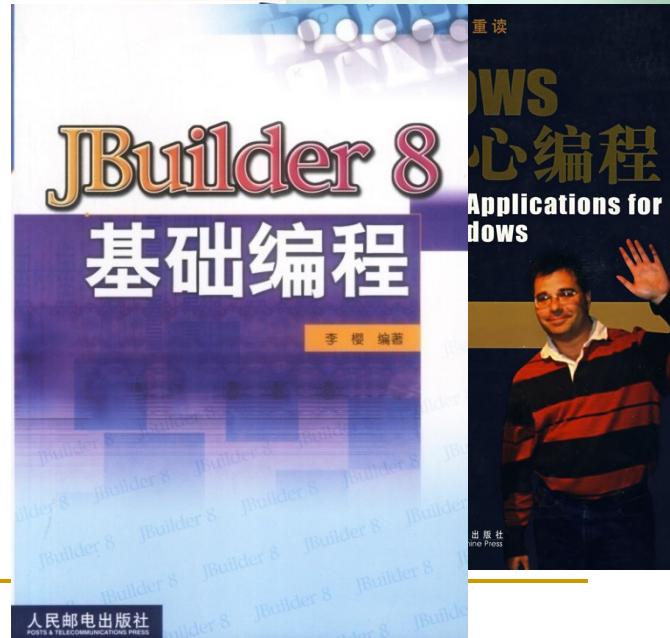




2.2 软件工程背景和意义

什么是程序 (2/2) ?

- 良好、严格语法和语义
- 主要类型
 - 面向机器：如汇编语言、机器语言等
 - 面向过程：如 Fortran, Pascal, C 等等
 - 面向对象：如 Java 等等
 - 面向问题：如结构化查询语言 SQL 等等

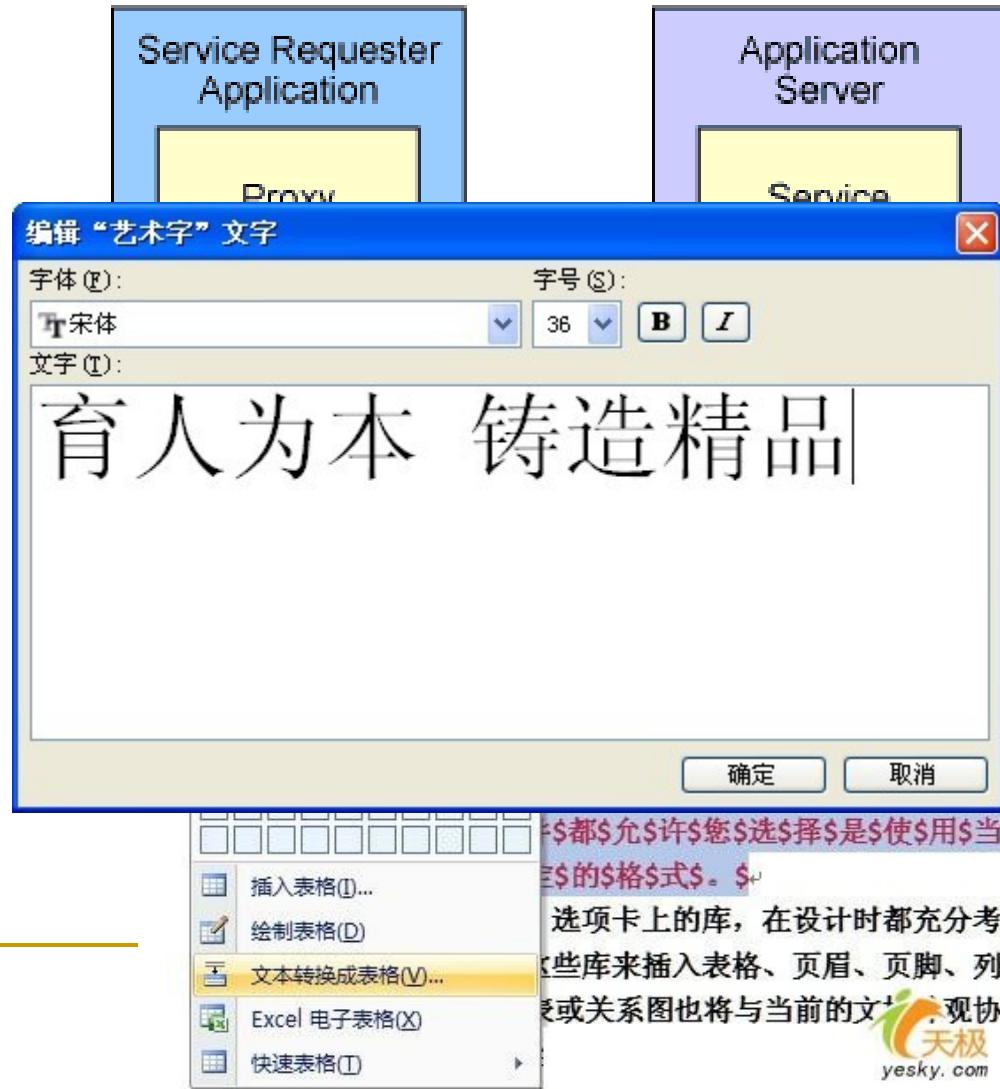




2.3 软件工程背景和意义

什么是文档？

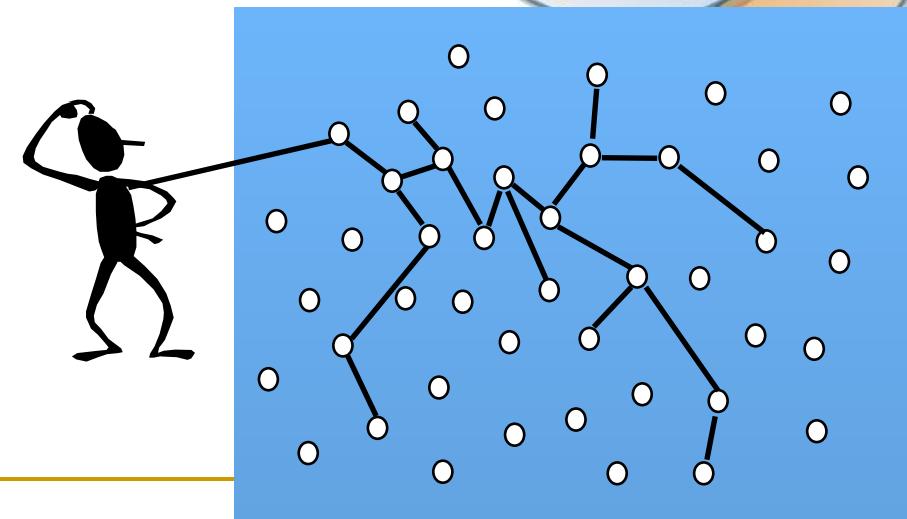
- 记录软件开发活动和阶段性成果、理解软件所必需的阐述性资料
 - 需求分析文档
 - 软件设计文档等
- 编写文档目的
 - 促进开发 / 管理 / 维护；
 - 便于人员（用户，开发人员）交流



2.4 软件工程背景和意义

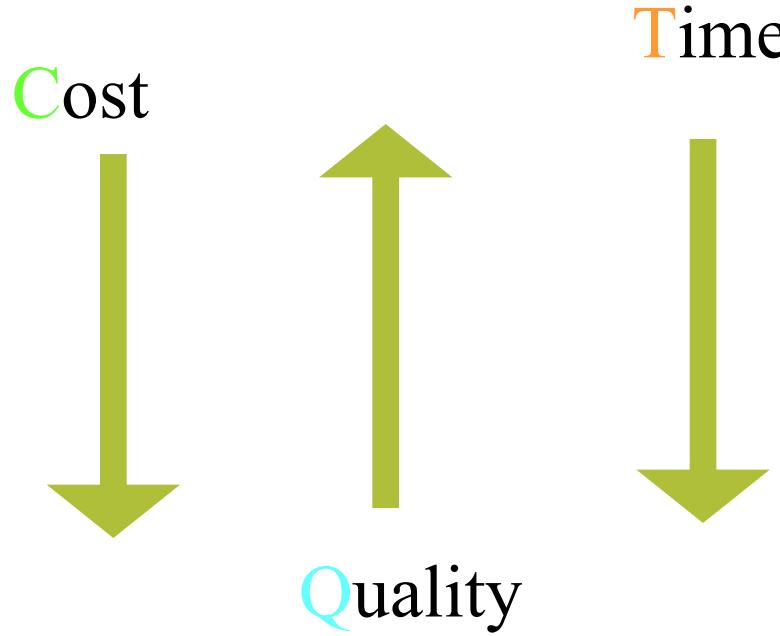
软件特点

- 不老化
- 逻辑产品
- 复杂
- 生产只需复制
- 成本、进度难估计
- 依赖于人



软件危机

Crisis in Software Industry



*Choose
tow ?*



软件危机

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。

软件危机的典型表现

- 1 对软件开发成本和进度的估计常常很不准确
- 2 用户对“已完成的”软件系统不满意的现象经常发生
- 3 软件产品的质量往往靠不住
- 4 软件常常是不可维护的

软件危机的典型表现

4 软件通常没有适当的文档资料

5 软件成本在计算机系统总成本中所占的比例逐年上升

6 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

1.1.2 产生软件危机的原因

与软件本身特点有关

1 软件不同于硬件，管理和控制软件开发过程相当困难。

2 如果运行中发现了错误，很可能是因为遇到了一个在开发时期引入的在测试阶段没能检测出来的错误。

3 规模庞大，而且程序复杂性将随着程序规模的增加而呈指数上升。

与软件本身特点有关

4 对用户要求没有完整准确的认识就匆忙着手编写程序是许多软件开发工程失败的主要原因之一。

5 相当多的软件专业人员对软件开发和维护还有不少糊涂观念。在实践过程中或多或少地采用了错误的方法和技术：主要原因。

6 错误的认识和做法主要表现为忽视软件需求分析的重要性，认为软件开发就是写程序并设法使之运行，轻视软件维护等

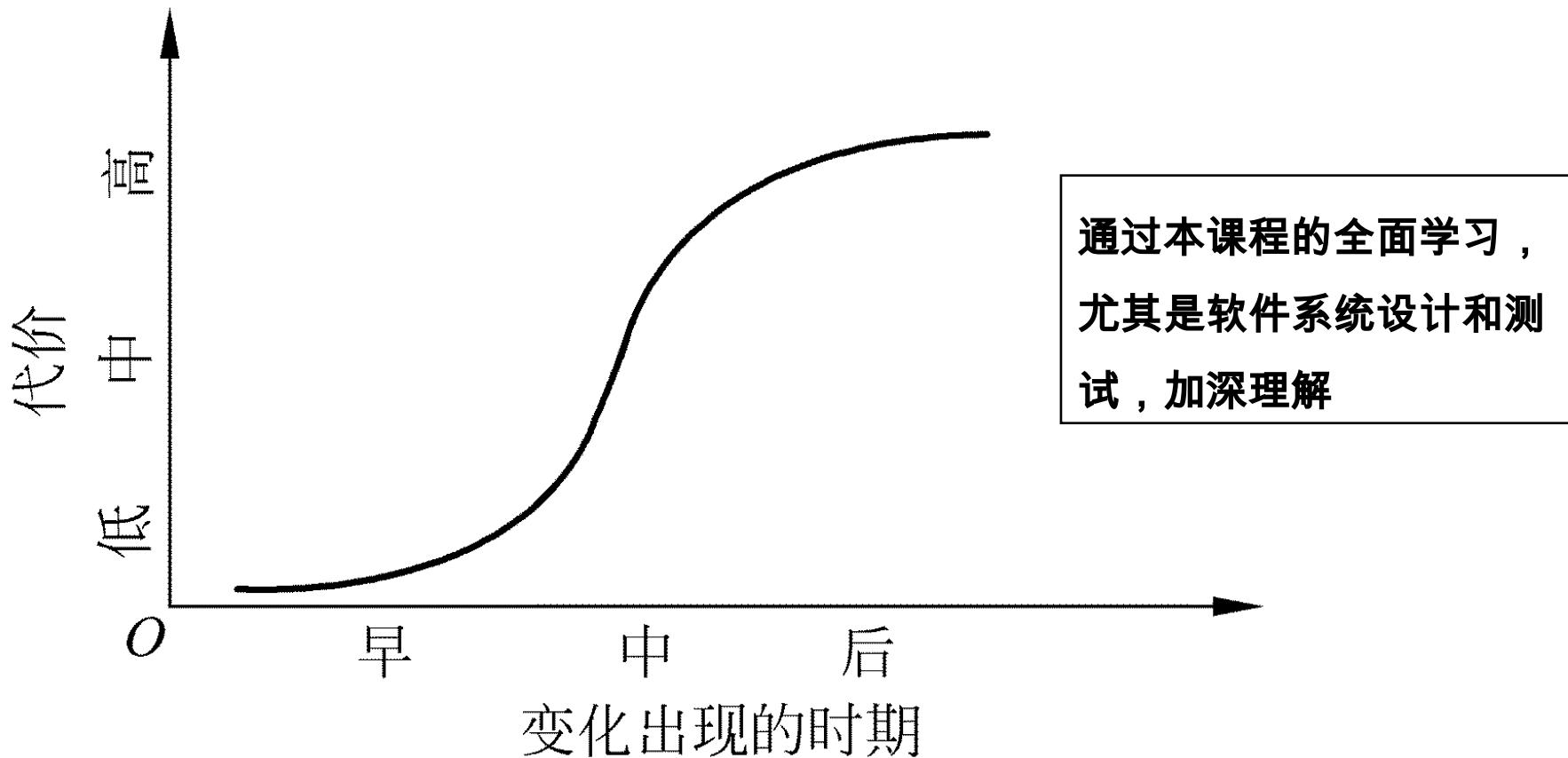
软件开发与维护的方法不正确有关

1只重视程序而忽
视软件配置其余
成分的糊涂观念。

2在定义时期没有
正确全面地理解
用户需求，直到
测试阶段或软件
交付使用后才发
现“已完成的”
软件不完全符合
用户的需要。

3严重的问题是在
软件开发的不同
阶段进行修改需
要付出的代价是
很不相同的

在软件开发的不同阶段进行修改需要付出的代价



1.1.3 消除软件危机的途径

1 首先应该对计算机软件有一个正确的认识。

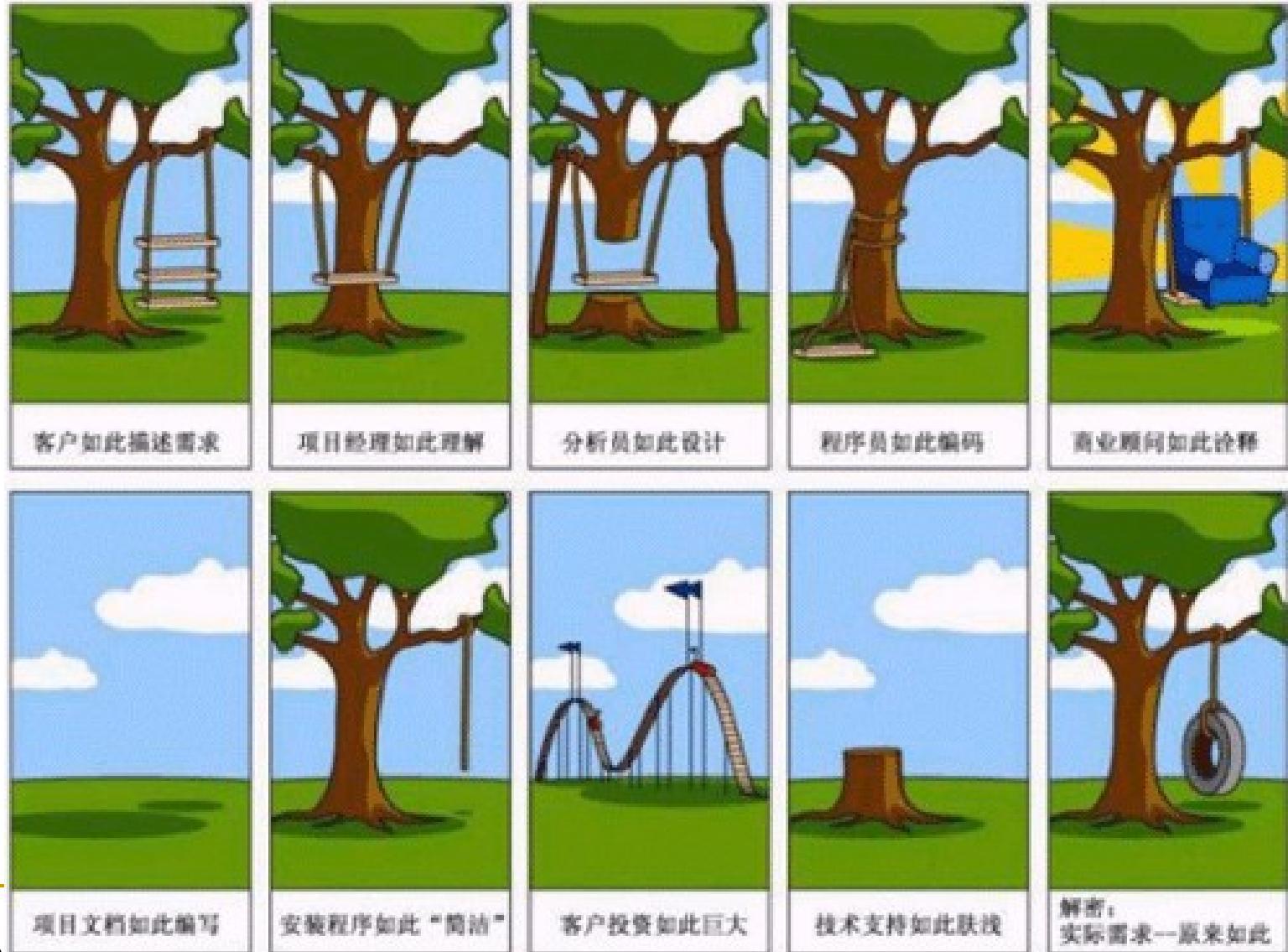
2 软件开发不是某种个体劳动的神秘技巧，而应该是各类人员协同配合，共同完成的工程项目。

3 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方法。

4 应该开发和使用更好的软件工具。

2.4 软件工程背景和意义

什么是软件危机 – 开发维护中遇到的一系列问题

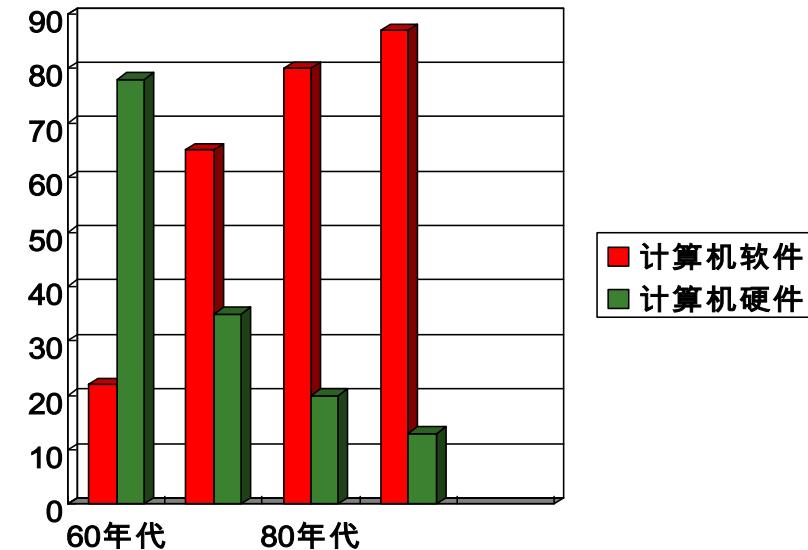


2.4 软件工程背景和意义

软件危机的表现 (1 of 3)

■ 成本高

- IBM 360 OS, 5000 多人年 , 耗时 4 年 (1963 - 1966) , 花费 **2亿多美元**
- 美国空军 : 1955 年软件占总费用 (计算机系统) 的 18% , 70 年 60% , **85 年达到 85 %**
- 美国全球军事指挥控制系统 , 硬件 1 亿美元 , **软件高达 7.2 亿美元**
- 计算机软件和硬件费用比



2.4 软件工程背景和意义

软件危机的定义

■ 软件质量得不到保证

- 软件应用面的扩大
企业管理、办公、金融、航天、军事等
- 80 年代欧洲亚丽士公司生产的软件，性能达不到要求，造成企业控制、生产瘫痪
- 美国阿托拉斯火箭发射失败
- 英国 1986 年开发的“潘多拉”系统，由于软件性能达不到要求，1989 年 1 月 1 日系统无法启动，造成 3 年下马
- 日本第 5 代机因为软件问题，3 年下马

■ 由于质量问题造成的灾难

- 20101029, USA



2.4 软件工程背景和意义

软件危机的表现 (3 of 3)

- 进度难以控制
 - Time --High
 - Cost -- High
- 维护非常困难



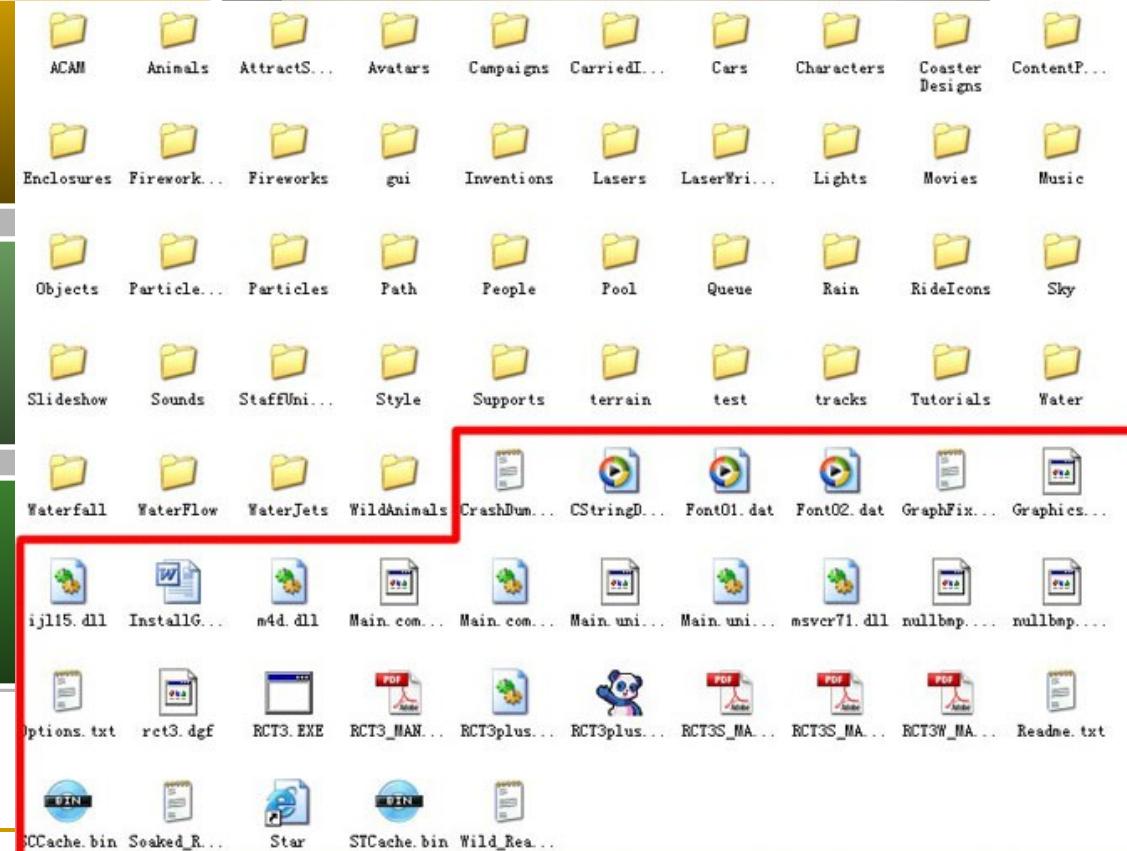
2.4 软件工程背景和意义

产生软件危机的根源

逻辑产品，复杂性高

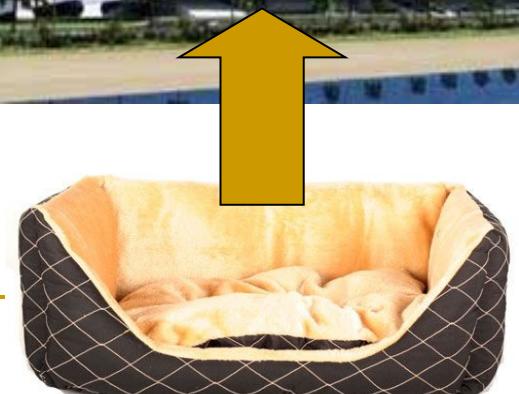
规模大

缺乏有效工具



2.4 软件工程背景和意义 对软件开发的深层次认识

- 软件系统 vs 程序 ==
不一样
- 须按工程化的方法组
织生产和管理，须经
过分析、设计、实现
、测试、维护一系列
过程和活动



2.4 软件工程背景和意义

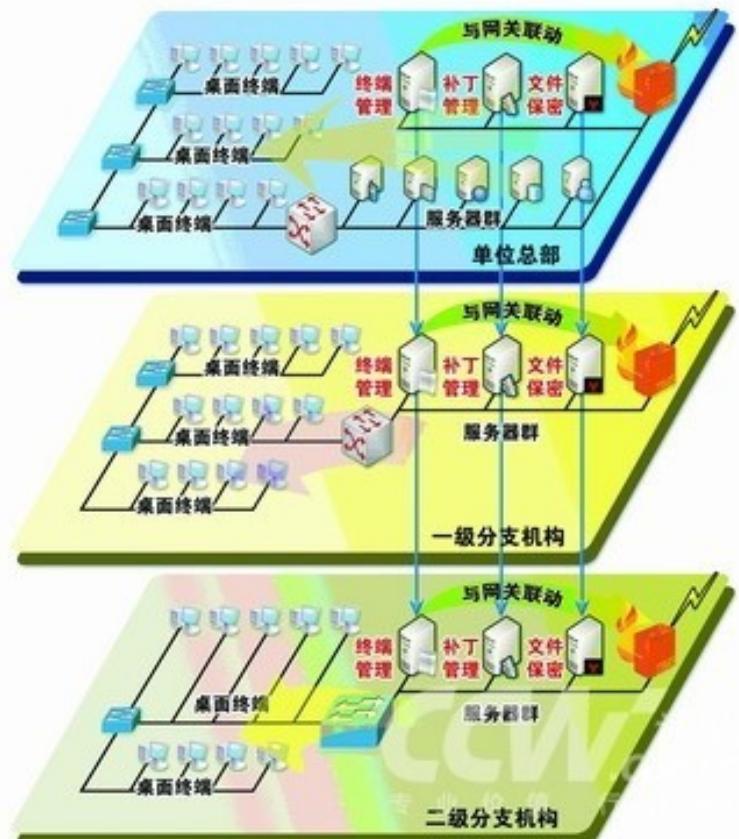
解决危机的技术途径

- 20th-60 末年代爆发
- 提出有效的方法和工具支持软件开发
 - 1968 年软件工程（ SE ）概念和思想
 - 20th-70 ， SA 开发方法
 - 20th-80 ， OO 开发方法
 - 新的技术：重用、快速原型、需求工程
 - 典型技术 : COM, Java, C++, J2EE, .Net,
 - 支撑工具和环境 : Eclipse, MS 2008, WebLogic, ...

2.4 软件工程背景和意义

解决危机的管理途径

- 20th-90，软件危机更为严重，
问题出在哪里？
- 20th-80末，美国 DoD 和工业
界开始认识到管理的重要性
 - 美国 DoD 的研究，70% 由
于管理不善导致项目失败
 - 管理 --- 影响软件项目成功开
发的全局性因素，而技术只
影响局部



2.5 软件工程背景和意义

软件工程 (Software Engineering, SE)

- 软件危机 --> 根源 --> 解决途径 : 软件工程



软件已经成为社会基础设施之一

----2000 美国科学院给 B Clinton 总统的报告

- 用户很不满意
- 质量很不可靠
- 软件成本上升
- 供不应求
- 质量下降
- 典型案例

运用工程的、数学的、计算机等科学概念、方法和原理来指导软件开发和管理和维护的一门学科

- 产生 : 1968 年 NATO 计算机科学会议

软件工程

1968 年在第一届 NATO 会议上曾经给出了软件工程的一个早期定义：“软件工程就是为了经济地获得可靠的且能在实际机器上有效地运行的软件，而建立和使用完善的工程原理。”

1993 年 IEEE 进一步给出了一个更全面更具体的定义：“软件工程是：①把系统的、规范的、可度量的途径应用于软件开发、运行和维护过程，也就是把工程应用于软件；②研究①中提到的途径。

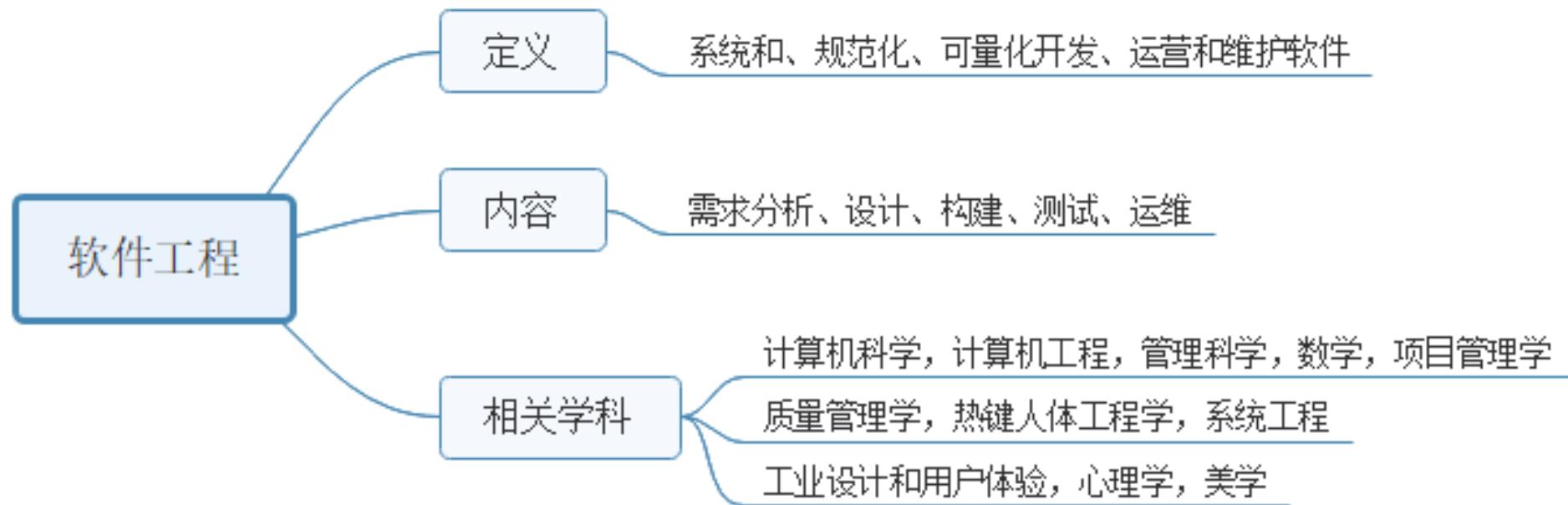
对比造飞机	软件	成功 / 失败的结局会怎样
玩具阶段	Hello world	只限于自己
爱好者	用 JavaScript , Asp .net , Ruby 写写 网站	基本限于自己
先行者、探索者 ：莱特兄弟	专研并应用新技术 在产品中	很多失败，但是为后来 者奠定了基础
成熟的工业： 制造	MIS , B2B , E B , . . .	影响一个公司、行 业、. . . 一个重要的失败可能导 致很多人失去工作

分析

- 如果一架✈️上有一个功能，使用概率不到万分之一，你要做吗：
 1. 根本不考虑
 2. 如果没时间实现，就算了
 3. 做了，但是不告诉客户
 4. 做了，而且不厌其烦的告诉客户

→ 安全功能

- (爆炸 、 引擎故障、 飞鸟撞击。)
- 能否在万米高空，面对突然出现的异常，程序正在高速运行，是否能依然正常工作、安全退出，并保证整个数据不被破坏？
- 我们未来去面试的时候，说，我懂 JAVA ， C++ ，写了很多程序，做了很多项目，懂软件工程，写了很多文档，然而，一个合格的软件工程师到底是什么？



软件的特殊性

复杂性：百万行代码、万个文件

不可见性：看不到源代码具体怎样运行

易变性：人们期待软件做新的事情、适应新的硬件

服从性：服从硬件、客户、行业（例如银行利率）

非连续性：输入很小，输出很大

Fred Brook Jr. No Silver Billet

Vaclav Rajlich

计算机科学 vs 软件工程

计算机科学	软件工程
发现和研究长期的客观真理	构建短期的实际结果
理想化的	对各因素的折中
确定性，完美，通用性	对不确定性和风险的管理； 具体应用
各个分支学科独立深入研究； 做出成果	应用各个分支相关知识； 解决问题
统一的理论	百花齐放的实践
强调原创	最成熟的、最好的实践方法
形式化，追求简明的公式	在实践中建立起来的灵感
正确性	可靠性

- 历史证明，从工程角度出发，用“蛮力”提高计算速度的工程方法远远甩开了同时代的各种“智能”方案
- 大部分大学生毕业后投身于解决具体工程问题
 - 哲学家：我思，故我在
 - 科学家：我发现，故我在
 - 工程师：我构建，故我在

总结

- 软件危机产生的时间和背景
- 软件危机的现象
- 软件危机的原因
- 软件工程概念

- Build to learn
- Build to show
- Build to serve
- Build to win
- 不再纠结于科学和工程
- 在不同阶段、不同身份，投入到合适的项目类型中

- 作业 1: 用 20 分钟写一个自动生成小学四则运算的程序，并且，增加
 - 支持判断对错，累计分数，倒计时
 - 支持括号
 - 用户界面可以是中英文选择
 - 把上面功能移到一个网页上
 - 与同学比较各自功能异同

- 作业 2: 一周之内看完本课程一本教材，并且
 - 列出 5-10 个你看不懂的问题，表明出处
 - 列出资料，说明你的疑问
 - 说明你提问的原因
 - 模版：我看了一段文字。。。。有了一个问题。。。有这些说法。。。根据我的经验。。。我得到这些经验。。。但是还是不懂，我的困惑是。。。。

• • •

■ 作业 3: 有很多种类软件 , OS , Apps , Games , Web App , Tools :

- 你为何成为其用户 ?
- 他们目标盈利是什么 ?
- 他们怎么实现目标 (盈利) 的 ?
- 这些软件如何处理 Bug ? 如何更新版本 ?
- 同类软件之间如何竞争 ?

(备用) 作业 1st

1. 什么是软件危机 , 结合软件危机的现象进行阐述
2. 简述并分析产生软件危机的原因
3. 何为软件工程 , 并稍深入讨论你对此概念的理解
4. 谈谈你对本课程的内容 (讲什么) 、目标 (课程要求) 的理解
5. 结合上题 (教学目标) , 为了实现目标 , 你有哪些优势和哪些有待提升的 ? 你准备怎样做
用笔手写 , 下次课程交 (不是每位同学都交 , 有解释)

复习 / 预习

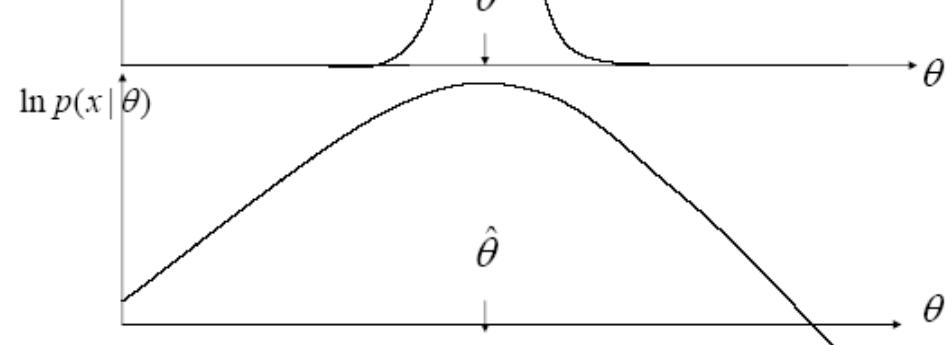
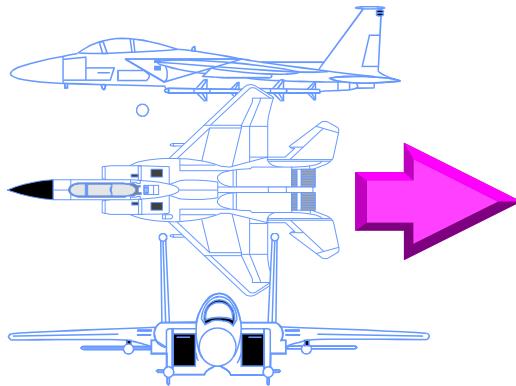
- JAVA /C++/PHP , My SQL, 自己搭建编程平台环境 (必做)
- 看教材 (Page?)
- UML (选做)
- Rational Rose 2003 (选做)
- 建立 Project team , 选出 PM (Project Manager) , 基本了解本课程 project 的大致要求和内容 (必做 , 详见后文)

Assignments

- Preparation for the Lab works
 - 3-5 persons per team and one PM
 - Analysis/design/implementing a software project, without any limitation on coding language (platform), database system used, or the function of the project
 - Meeting every week within a team
 - PM meeting every two week in the experimental classes

3.1 What Is a Model?

- A model is a simplification of reality
 - 现实事物的抽象反映



$$p(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty$$

What is a model?

- **Grady Booch, chief scientist at Rational Software**
 - **Provides the blueprints of a system**
 - **May encompass detailed plans, as well as more general plans that give a view of the system under construction**
 - **A good model includes those elements that are not relevant to the given level of abstraction**
 - **Every system may be described from different aspects using different models, and each model is therefore a semantically closed abstraction of the system**
 - **A model may be structural, emphasizing the organization of the system, or it may be behavioral, emphasizing the dynamics of the system**

3.2 Why Model?

你在学习和工作中
哪些地方需要建模

？

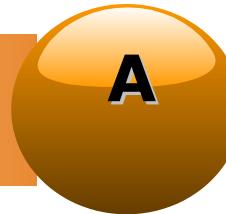
什么模型？

- Modeling achieves four aims:
 - To visualize a system as you want it to be
 - (可视化系统)
 - To specify the structure or behavior of a system
 - (详细描述系统的结构和功能)
 - Gives a template guiding you in constructing
 - (提供实现系统的模板)
 - Documents the decisions
 - (写出高质量的文档)
- You build models of complex systems because you cannot comprehend such a system in its entirety and to better understand the system you are developing.

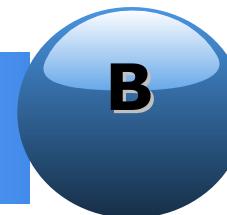


3.3 Software Teams Often Do Not Model

Start coding from requirement



Work longer & create more codes

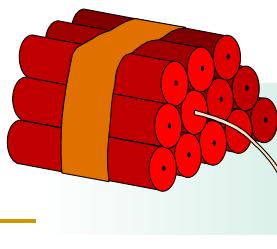
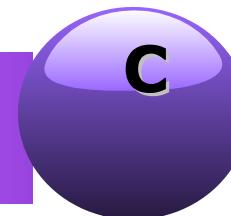


Modeling-

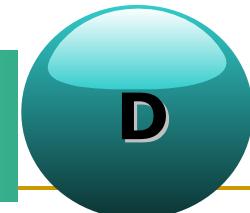
a common
thread to
successful
projects

-- 成功的保证 !

Lacks any planned architecture



Doomed to failure!!!



3.4 Four Principles of Modeling

Principle 1: The Choice of Model Is Important

The models you create profoundly influence how a problem is attacked and how a solution is shaped.

□ (选择合适的模型)



$$p(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty$$

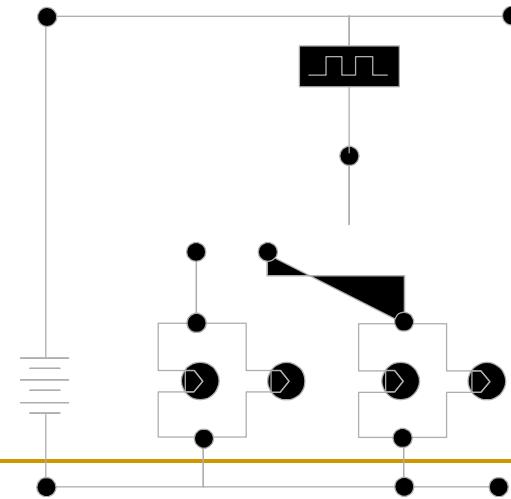
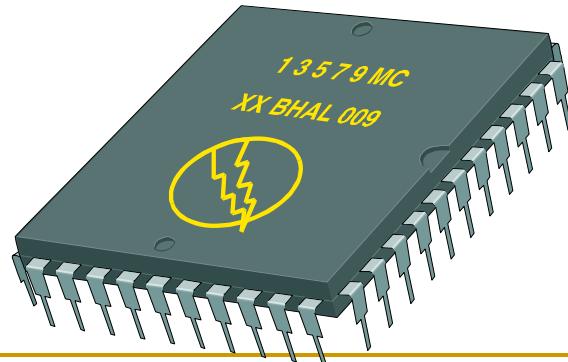
- **The right models illuminate the most difficult development problems, offering insight that you could not gain otherwise.**
- **The wrong models mislead you, causing you to focus on irrelevant issues.**
- **In software, the models you choose can greatly affect your world view.**
- **If you build a system through the eyes of a database developer, you'll likely end up with entity-relationship models that push behavior into stored procedures and triggers.**

- **If you build a system through the eyes of an object-oriented developer, you'll end up with a system that has its architecture centered around many classes and patterns of interaction that direct how those classes work together.**
- **Each world view leads to a different kind of system with different costs and benefits.**
 - ***The Unified Modeling Language User Guide, Booch, 1999***

3.4 Four Principles of Modeling

Principle 2: Levels of Precision May Differ

- Every model may be expressed at different levels of precision.
(模型的多层次) (抽象层次)
 - depending on:
 - Who is viewing the model. (给谁看 ?)
 - Why they need to view it (为什么要看 ?)



- **If you are building computer chips, sometimes you need a 30,000-foot view. For example, you need your investors to visualize the end product. Other times, you need to get down to the level of the circuits.**
- **When developing a GUI system, a quick and dirty executable model of the user interface may be all you need to communicate your intentions.**
- **Other times, when you are dealing with cross-system interfaces of network bottlenecks, you need to model down to the bit level.**
- **In either case, the best models are those that let you choose your degree of detail, depending on who is doing the viewing and why they need to view it.**

Booch, 1999

3.4 Four Principles of Modeling

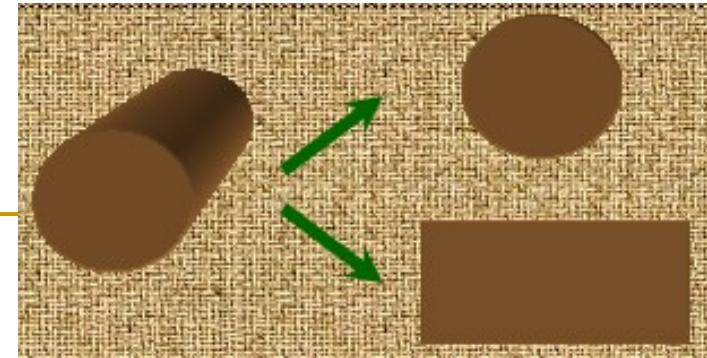
Principle 3: Connected to Reality

- All models simplify reality & reflects potentially fatal characteristics
 - 如实反映问题、且使问题容易解决

软件工程方法学家 Grandy Booch:

揭示对象的本质特征，将该对象与其它对象区别开来，从而清晰定义概念的边界。

---*Object-Oriented Analysis and Design with Application*



- A physical model of a building that doesn't respond the same way as the real materials has limited value. It's best to have models that have a clear connection to reality.
- Where that connection is weak, you need to know exactly how those models are divorced from the real world.
- All models simplify reality. The trick is to be sure that your simplifications don't mask any important details. A good model reveals any potentially fatal flaws in design. **Booch, 1999.**

3.4 Four Principles of Modeling

Principle 4: No Single Model Is Sufficient

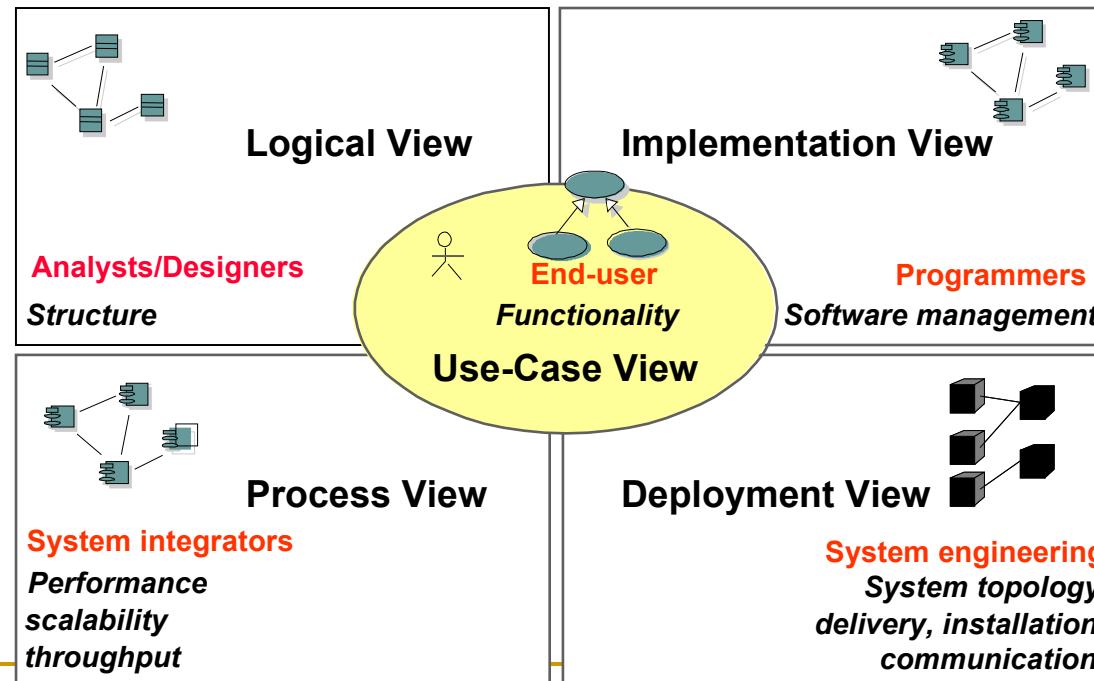
- No single model is sufficient. Every non-trivial system is best approached through a small set of nearly independent models.
 - 模型的多角度



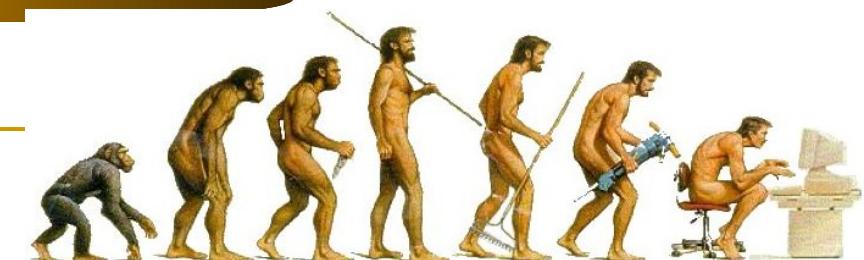
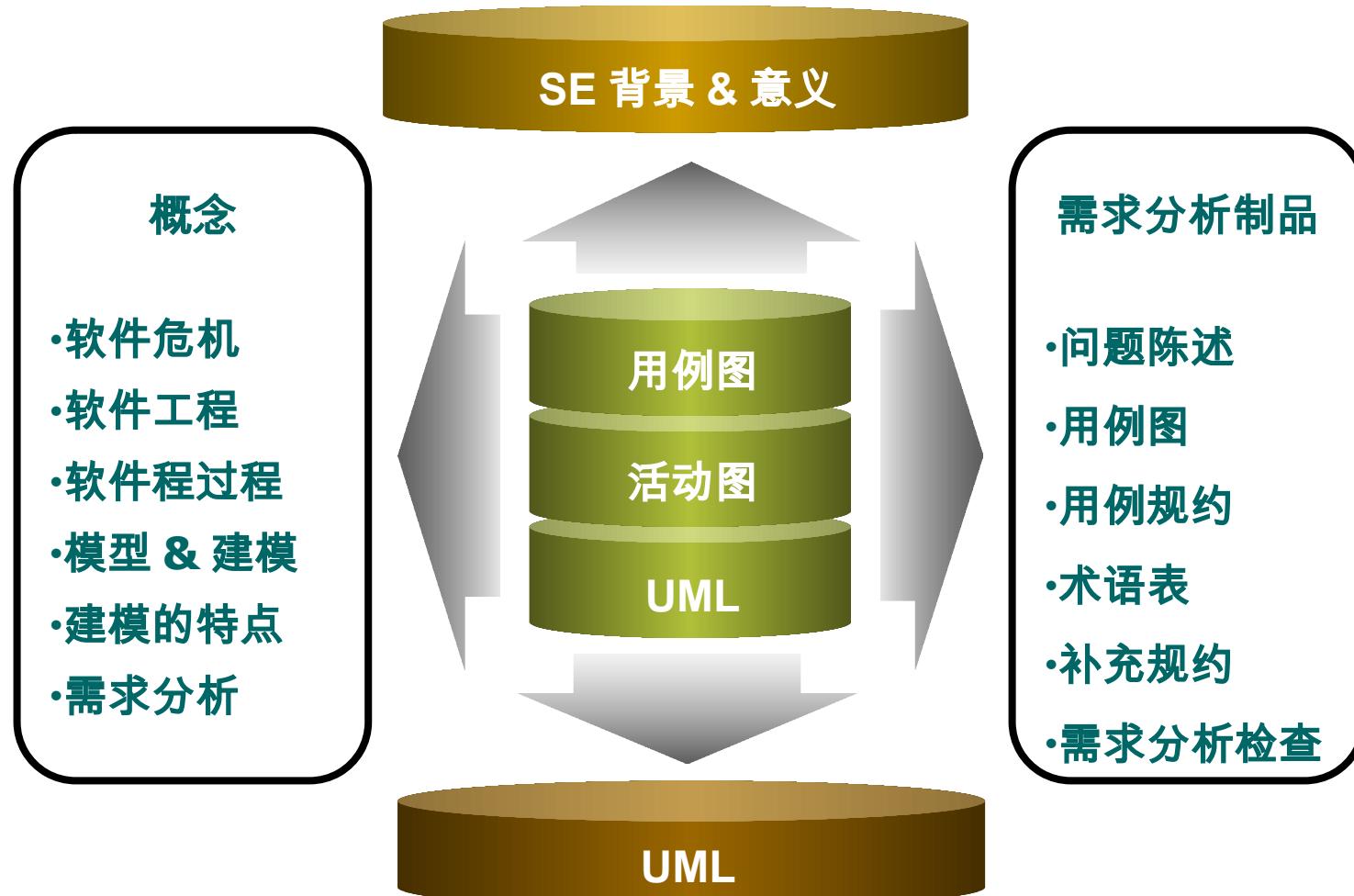
- **The right models illuminate the most difficult development problems, offering insight that you could not gain otherwise. The wrong models mislead you, causing you to focus on irrelevant issues.**
- **In software, the models you choose can greatly affect your world view. If you build a system through the eyes of a database developer, you'll likely end up with entity-relationship models that push behavior into stored procedures and triggers.**
- **Booch, 1999.**

软件分析与设计中的多层次、多角度建模

- No single model is sufficient. Every non-trivial system is best approached through a small set of nearly independent models.
 - 模型的多角度



Summary



练习和讨论

1. 什么是建模（Modeloing）
2. 一个好的模型（建模）的四项基本原则
3. 很多 IT 人喜欢上 BBS，目前有名的 BBS 有几个？它们为什么能后来居上？